

PCT

世界知的所有権機関
国際事務局
特許協力条約に基づいて公開された国際出願

3101327^号

<p>(51) 国際特許分類 G06F 9/44</p>	<p>A1</p>	<p>(11) 国際公開番号 WO98/45776</p> <p>(43) 国際公開日 1998年10月15日(15.10.98)</p>
<p>(21) 国際出願番号 PCT/JP97/04864</p> <p>(22) 国際出願日 1997年12月25日(25.12.97)</p> <p>(30) 優先権データ 特願平9/92091 1997年4月10日(10.04.97) JP</p> <p>(71) 出願人 (米国を除くすべての指定国について) インターナショナル・ビジネス・マシーンス・コーポレーション(INTERNATIONAL BUSINESS MACHINES CORPORATION)[US/US] 10504、ニューヨーク州アーモンク New York, (US)</p> <p>(72) 発明者 ; および (75) 発明者 / 出願人 (米国についてのみ) 南 和宏(MINAMI, Kazuhiro)[JP/JP] 鈴木俊宏(SUZUKI, Toshihiro)[JP/JP] 〒242 神奈川県大和市下鶴間1623番地14 日本アイ・ビー・エム株式会社 大和事業所内 Kanagawa, (JP)</p> <p>(74) 代理人 弁理士 坂口 博, 外(SAKAGUCHI, Hiroshi et al.) 〒242 神奈川県大和市下鶴間1623番地14 日本アイ・ビー・エム株式会社 大和事業所内 Kanagawa, (JP)</p>		<p>(81) 指定国 BR, CA, CN, CZ, HU, ID, JP, KR, MX, PL, RU, SG, SK, US, 欧州特許 (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).</p> <p>添付公開書類 国際調査報告書</p>
<p>(54) Title: MOVING OBJECT, MOVING OBJECT CONTROL METHOD, METHOD OF AND APPARATUS FOR GENERATING MOVING OBJECTS, AND STORAGE MEDIUM FOR PROGRAM FOR GENERATING MOVING OBJECTS</p> <p>(54) 発明の名称 移動オブジェクト、移動オブジェクトの制御方法、移動オブジェクト群の生成方法・生成装置、移動オブジェクト群を生成するプログラムを格納する記憶媒体</p> <p>(57) Abstract A template in which complicated behaviors of a moving agent are classified into basic behavior patterns is provided. This template is linked with activities of the moving agent while the moving agent exists at a place before it moves and activities executed after the moving agent has moved. The template defines instructions executed before and after the moving agent moves, in accordance with the kind thereof, and sends out the instructions to the moving agent. The moving agent carries out in response thereto generation of a clone, transfer of the same to another place and fusion of clones. The plan-object held in the moving agent defines the order of execution of the template and activities. A moving agent capable of carrying out a process in a short period of time efficiently, and capable of suppressing the occupation of other resources to a minimum level can be provided.</p> <div data-bbox="722 1260 1356 1638"> <p>The diagram illustrates a system for managing moving objects. It includes a client (101) connected to a server (102). A monitoring agent (107) monitors events, which are then processed by a moderator agent (111) and a planner agent (113). These agents interact with a moderator template (160) and various other templates (114, 123, 125, 127, 129, 131, 133, 135, 137, 139, 141, 143, 145, 147, 149, 151, 153, 155, 157, 159, 161, 163, 165, 167, 169, 171, 173, 175, 177, 179, 181, 183, 185, 187, 189, 191, 193, 195, 197, 199). The system also includes a plan-object (111) and a plan-object (113). The diagram shows the flow of information and control between these components.</p> </div> <p>101 ... client 102 ... server 107 ... monitoring an event 111 ... moderator agent 113 ... planner agent 114 ... combining the results 123 ... actor agent 125 ... actor agent 160 ... moderator template A ... delivering a plan B ... request C ... crowd</p>		

本発明は、移動エージェントの複雑な行動を基本的行動パターンに分類したテンプレートを提供する。このテンプレートは、移動前のプレースにおいて実行されるアクティビティと移動後に実行されるアクティビティにリンクしている。テンプレートは、その種類に応じ、移動前と移動後において実行するインストラクションを定義しており、これを移動エージェントに送出する。移動エージェントはこれに応答して、クローンの作成、他のプレースへの移動、クローン同士の融合等を行う。また、移動エージェントに保持されているプラン・オブジェクトは、テンプレートとアクティビティの実行順序を定義している。本発明によれば、効率的かつ、処理に要する時間が短かく、他の資源の占有を最小限に抑えることのできる移動エージェントを提供することができる。

PCTに基づいて公開される国際出願のパンフレット第一頁に掲載されたPCT加盟国を同定するために使用されるコード(参考情報)

AL アルバニア	FI フィンランド	LR リベリア	SK スロヴァキア
AM アルメニア	FR フランス	LS レント	SL シェラ・レオネ
AT オーストリア	GA ガボン	LT リトアニア	SN セネガル
AU オーストラリア	GB 英国	LU ルクセンブルグ	SZ スワジランド
AZ アゼルバイジャン	GD グレナダ	LV ラトヴィア	TD チャード
BA ボスニア・ヘルツェゴビナ	GE グルジア	MC モナコ	TG トーゴ
BB バルバドス	GH ガーナ	MD モルドヴァ	TJ タジキスタン
BE ベルギー	GM ガンビア	MG マダガスカル	TM トルクメニスタン
BF ブルキナ・ファソ	GN ギニア	MK マケドニア旧ユーゴスラヴィア	TR トルコ
BG ブルガリア	GW ギニア・ビサウ	共和国	TT トリニダード・トバゴ
BJ ベナン	GR ギリシャ	ML マリ	UA ウクライナ
BR ブラジル	HR クロアチア	MN モンゴル	UG ウガンダ
BY ベラルーシ	HU ハンガリー	MR モーリタニア	US 米国
CA カナダ	ID インドネシア	MW マラウイ	UZ ウズベキスタン
CF 中央アフリカ	IE アイルランド	MX メキシコ	VN ヴィエトナム
CG コンゴ	IL イスラエル	NE ニジェール	YU ユーゴスラビア
CH スイス	IS アイスランド	NL オランダ	ZW ジンバブエ
CI コートジボアール	IT イタリア	NO ノールウェー	
CM カメルーン	JP 日本	NZ ニュー・ジーランド	
CN 中国	KE ケニア	PL ポーランド	
CU キューバ	KG キルギスタン	PT ポルトガル	
CY キプロス	KP 北朝鮮	RO ルーマニア	
CZ チェッコ	KR 韓国	RU ロシア	
DE ドイツ	KZ カザフスタン	SD スーダン	
DK デンマーク	LC セントルシア	SE スウェーデン	
EE エストニア	LI リヒテンシュタイン	SG シンガポール	
ES スペイン			

明細書

移動オブジェクト、移動オブジェクトの制御方法、移動オブジェクト群の生成方法・生成装置、移動オブジェクト群を生成するプログラムを格納する記憶媒体

5 [技術分野]

この発明は、コンピュータネットワークにおけるデータ処理方法に関し、より詳しくは、分散コンピュータ環境あって、ネットワーク上に存在する遠隔サーバに移動し、該遠隔サーバにおいてインストラクションを形成する移動エージェント技術に関する。

[背景技術]

従来より、分散コンピュータ環境において、ネットワーク上に存在するサーバに移動し、インストラクションを形成する移動エージェント（モバイルエージェント）技術が存在する（特開平7-182174号公報(USP5,603,031)、「日経コミュニケーション別冊、最新インターネットテクノロジー」日経BP社pp104-117、西田 文比古、藤原進 著等）。

かかる移動エージェントは、「移動エージェント」と「プレース」という2つの基本要素を基に構成される。この移動エージェントは、自分自身の内部状態を保持したままネットワーク上に存在するプレース間を移動することができる。そして、このプレースにおいて他のエージェント（移動エージェントまたは非移動エージェント）とコンタクトし、必要なサービスを受けることができる。プレースは、ネットワーク上に存在するサーバによって提供されているエージェントの移動場所であり、

エージェント間のコンタクトをサポートし、また、ハードウェア及びプラットフォームの違いを吸収している。

かかる移動エージェントの技術は、社内会議のセッティングを社員のスケジュールや会議室の予約状況に応じてダイナミックに調整したり、
5 ネットワーク上に分散する所望の情報を入手する等、これまで人手によってなされていた作業をこの移動エージェントが代行することを可能にしている。

しかし、このような複数人のスケジュールの空きを探し、会議室を予約する等の処理を従来の移動エージェントが行う場合、複数人のスケジュールの情報の存在するプレースや会議室の予約情報の存在するプレースをエージェントが順々に渡り歩き、出席者の空き時間や会議室の空き時間を調査し、出席者の空き時間や会議室の空き時間を仮予約し、最終的に共通の時間を探し出し、都合のいい時間が存在した場合には、仮予約を本予約として出席者に知らせ、会議室の予約を行う必要がある。

15 しかし、出席者の数や利用可能な会議室の数が多い場合、仮予約と本予約の間に時間差が生じ、他の社内会議の予約のためのエージェントが待たされたり、利用可能として認識できる時間帯が少ないものになってしまう可能性もあった。かかる場合、複数のエージェントが並列に作業を進めれば、処理時間も短縮でき、移動エージェントがロックする仮予約の時間も短縮することができる。また、実行する処理が複雑で並列に
20 処理する必要がある場合には、クライアントサイトから複数の移動オブジェクトを放ち、順番に作業させ、その結果を再びクライアントサイトに送り、その結果を人手によって所望の処理を加える必要があった。

かかる移動エージェントの並列処理の要求に対し、従来型の J a v a
25 ベースのモバイル・エージェント ("J a v a" はサンマイクロシステムズ社の商標) は、ネットワークを移動する際、プロセスのステートを

保持したまま移動する機能をサポートしていないため、移動エージェントを処理途中で分岐させ、並列に処理を進めていくことはできない。また、テレスクリプトの移動エージェント（"テレスクリプト"は、ジェネラルマジック社の商標）は、ネットワークを移動する際、プロセスのステートを保持したまま移動することができるが、アクティビティの実行順序等の管理情報を保持したまま移動する機能はサポートされていないため、移動エージェントを分割し、並列処理することはできない。

また、並列処理を行わない通常のシリアル処理を行う移動エージェントの場合であっても、その作成や運用は、行き先やエージェントの振る舞いを定義するためのプログラミング必要である。このため、プログラマに高度なプログラミングの知識が要求され、またその開発には多大な時間と労力が必要となっていた。

[発明の開示]

本発明の一つの目的は、実行時において、他の資源の占有を最小限に抑えることのできる移動エージェントを提供することにある。

本発明の他の一つの目的は、効率的な処理を可能とする移動エージェントを提供することにある。

本発明の他の一つの目的は、移動エージェントを放ってから結果が返ってくるまでに要する処理時間を短縮することにある。

本発明の他の一つの目的は、高度なプログラミングの知識を必要とせず、移動エージェントの開発を可能にすることにある。

本発明の他の一つの目的は、移動エージェントの開発労力及び開発時間を可能な限り軽減することにある。

本発明の他の一つの目的は、開発者が直感的にその機能を把握でき、

親しみやすい移動エージェント開発用プログラム部品を提供することにある。

5 本発明は、移動エージェントの複雑な行動を基本的行動パターンに分類し、その組合せによって、移動エージェントの所望の行動パターンを定義できるテンプレート（モデレータ・テンプレート）を提供する。このテンプレートは、移動エージェントが渡り歩く各ブレースにおいてリクエストを出すアクティビティを、プレデセッサ（前任者：前のブレースで仕事をするアクティビティ）とサクセッサ（承継者：次のブレース
10 で仕事をするアクティビティ）として管理している。

テンプレートは移動エージェントの移動の態様により、Chain, AND Split, AND Join等複数種類用意されており、この種類に応じ前のブレース（オリジン）と次のブレース（ディスティネーション）において、移動エージェントにインストラクションを送出する。
15 移動エージェントは、このインストラクションに応答して、クローンの作成や、他のブレースへの移動、クローン同士の融合等を行う。

本発明の一態様においては、

20 (a) 前記第1のアクティビティとのリンク情報を管理するプレデセッサ・リストと、

(b) 前記第2のアクティビティとのリンク情報を管理するサクセッサ・リストと、

25 (c) 前記第1のブレースにおいて、前記移動モデレータ・エージェントに前記第2のブレースへの移動を要求するモデレーション・モジュールとを含む移動モデレータ・エージェントの制御の下、第1のブレースから第2のブレースへ移動し、前記第1のブレースにおいて第1のA

クティビティを実行し、前記第2のプレースにおいて第2のアクティビティを実行する移動オブジェクト群の一部を構成するオブジェクトが提供される。本発明の一態様において請求するオブジェクトは、本発明の好適な実施例におけるモデレータ・テンプレートに相当する概念である。

5

本発明の他の一態様においては、

(a) 前記第1のアクティビティとのリンク情報を管理するプレデセッサ・リストと、

10

(b) 前記第2のアクティビティとのリンク情報を管理するサクセッサ・リストと、

(c) 前記モデレータモジュールが前記第1のプレースに存在しているのか、前記第2のプレースに存在しているのを判断するステート管理部と、

15

(d) 前記第1のプレースにおいて、前記移動モデレータ・エージェントに前記第2のプレースへの移動を要求するモデレーション・モジュールとを含む移動モデレータ・エージェントの制御の下、第1のプレースから第2のプレースへ移動し、前記第1のプレースにおいて第1のアクティビティを実行し、前記第2のプレースにおいて第2のアクティビティを実行する移動オブジェクト群の一部を構成するオブジェクトが提供される。本発明の他の一態様において請求するオブジェクトは、本発明の好適な実施例におけるモデレータ・テンプレートに相当する概念である。

20

25

本発明の他の一態様においては、

(a-1) 前記第1のアクティビティとのリンク情報を管理するプ

レデセッサ・リストと、

(a-2) 前記第2のアクティビティとのリンク情報を管理するサ
クセッサ・リストと、

5 (a-3) 前記モデレータモジュールが前記第1のプレースに存在
しているのか、前記第2のプレースに存在しているのかを判断するステ
ート管理部と、

(a-4) 前記第1のプレースにおいて、前記移動モデレータ・エ
ージェントに前記第2のプレースへ前記移動オブジェクト群の移動を要
求するモデレーション・モジュールと、

10 を含むモデレータ・テンプレートと、

(b) 前記第1のアクティビティ、前記第2のアクティビティのいず
れが現在実行権限を有しているかを判断するプラン・イタレータと、

(c) 前記第1のアクティビティ、前記第2のアクティビティ、前記
15 モデレータ・テンプレートの実行順序を管理するプラン・オブジェクト
とを含む移動モデレータ・エージェントの制御の下、第1のプレースか
ら第2のプレースへ移動し、前記第1のプレースにおいて第1のアクテ
ィビティを実行し、前記第2のプレースにおいて第2のアクティビティ
を実行する移動オブジェクト群が提供される。

20 本発明の他の一態様においては、

(a) 前記移動モデレータ・エージェントに前記第1のアクティビテ
ィへのリンク情報を含む第1の移動モデレータ・エージェントと前記第
2のアクティビティへのリンク情報を含む第2の移動モデレータ・エー
ジェントの用意を要求する段階と、

25 (b) 前記第1のアドレスを入手する段階と、

(c) 前記第2のアドレスを入手する段階と、

(d) 前記第1の移動モデレータ・エージェントに前記第1のプレースに移動することを要求する段階と、

(e) 前記第2の移動モデレータ・エージェントに前記第2のプレースに移動することを要求する段階とを含む移動モデレータ・エージェントの制御の下、第1のアドレスによって特定される第1のプレースと第2のアドレスによって特定される第2のプレースへ移動し、前記第1のプレースにおいて第1のアクティビティを実行し、前記第2のプレースにおいて第2のアクティビティを実行する移動オブジェクト群を制御する方法が提供される。

本願明細書の特許請求の範囲において、第1の移動モデレータ・エージェントと第2の移動モデレータ・エージェントの用意を要求するとは、第1の移動モデレータ・エージェントと第2の移動モデレータ・エージェントに対応する2つのクローンを作成することのみならず、第1の移動モデレータ・エージェントまたは第2の移動モデレータ・エージェントに対応する1つのクローンが、第1のアドレスまたは第2のアドレスに移動し、現在処理中の移動モデレータ・エージェントが、第2のアドレスまたは第1のアドレスに移動する場合をも含む概念である。

本発明の他の一態様においては、

(a) 前記第1のプレースにおいて、前記移動モデレータ・エージェントに前記第2のアクティビティへのリンク情報を含む第1の移動モデレータ・エージェントと前記第3のアクティビティへのリンク情報を含む第2の移動モデレータ・エージェントの用意を要求する段階と、

(b) 前記第2のアドレスを入手する段階と、

(c) 前記第3のアドレスを入手する段階と、

(d) 前記第 1 の移動モデレータ・エージェントに、前記第 1 のプレースから前記第 2 のプレースに移動することを要求する段階と、

(e) 前記第 2 の移動モデレータ・エージェントに、前記第 1 のプレースから前記第 3 のプレースに移動することを要求する段階と、

5 (f) 前記第 2 のプレースにおいて前記第 2 のアクティビティを実行することを要求する段階と、

(g) 前記第 3 のプレースにおいて前記第 3 のアクティビティを実行することを要求する段階とを含む移動モデレータ・エージェントの制御の下、第 1 のアドレスによって特定される第 1 のプレースと第 2 のアドレスによって特定される第 2 のプレースと第 3 のアドレスによって特定される第 3 のプレースへ移動し、前記第 1 のプレースにおいて第 1 のアクティビティを実行し、前記第 2 のプレースにおいて第 2 のアクティビティを実行し、前記第 3 のプレースにおいて第 3 のアクティビティを実行する移動オブジェクト群を制御する方法が提供される。

10

15

本発明の他の一態様においては、

(a) 第 1 のプレースにおいて、第 1 の移動モデレータ・エージェントにリンクした第 1 のアクティビティを実行し、第 1 の処理結果を得る段階と、

20

(b) 前記第 1 のプレースに存在し、前記第 1 の処理結果を保持する前記第 1 の移動モデレータ・エージェントに対し第 2 のプレースに移動することを要求する段階と、

25

(c) 第 3 のプレースにおいて、第 2 の移動モデレータ・エージェントにリンクした第 2 のアクティビティを実行し、第 2 の処理結果を得る段階と、

(d) 前記第 3 のプレースに存在し、前記第 2 の処理結果を保持する

前記第 2 の移動モデレータ・エージェントに対し前記第 2 のプレースに移動することを要求する段階と、

5 (e) 前記第 2 のプレースにおいて、前記第 2 の移動モデレータ・エージェントに対し前記第 2 の処理結果を前記第 1 の移動モデレータ・エージェントに渡すことを要求する段階と、

10 (f) 前記第 2 のプレースにおいて、前記第 1 の移動モデレータ・エージェントが前記第 1 の処理結果とともに前記第 2 の処理結果を保持する段階とを含む移動モデレータ・エージェントの制御の下、複数のプレースを移動し、プレースに対応したアクティビティを実行する移動オブジェクト群を制御する方法が提供される。本願明細書の特許請求の範囲において、第 2 の移動モデレータ・エージェントの第 2 の処理結果を第 1 の移動モデレータ・エージェントに渡すとは、プランノードのコピーのみならず、Result の内容だけを渡すことも含む概念である。

15 本発明の他の一態様においては、

(a) 第 1 のプレースにおいて、エージェント識別情報を有する第 1 の移動モデレータ・エージェントにリンクした第 1 のアクティビティを実行し、第 1 の処理結果を得る段階と、

20 (b) 前記第 1 のプレースに存在し、前記第 1 の処理結果を保持する前記第 1 の移動モデレータ・エージェントに対し第 2 のプレースに移動することを要求する段階と、

25 (c) 第 3 のプレースにおいて、前記第 1 の移動モデレータ・エージェントの有するエージェント識別情報と同一の識別情報を有する第 2 の移動モデレータ・エージェントにリンクした第 2 のアクティビティを実行し、第 2 の処理結果を得る段階と、

(d) 前記第 3 のプレースに存在し、前記第 2 の処理結果を保持する

前記第 2 の移動モデレータ・エージェントに対し前記第 2 のプレースに移動することを要求する段階と、

(e) 前記第 2 のプレースにおいて、前記第 2 の移動モデレータ・エージェントに対し前記第 2 の処理結果を前記第 1 の移動モデレータ・エージェントに渡すことを要求する段階と、

(f) 前記第 2 のプレースにおいて、前記エージェント識別情報と同一の識別情報を有する移動モデレータ・エージェントを探索する段階と、

(g) 前記第 2 のプレースにおいて、前記第 1 の移動モデレータ・エージェントが前記第 1 の処理結果とともに前記第 2 の処理結果を保持する段階とを含む移動モデレータ・エージェントの制御の下、複数のプレースを移動し、プレースに対応したアクティビティを実行する移動オブジェクトを制御する方法が提供される。

本発明の他の一態様においては、

(a) 前記第 1 のアクティビティ、前記第 2 のアクティビティの実行順序を管理するプラン・オブジェクトと、

(b) 前記第 1 のアクティビティ、前記第 2 のアクティビティ、前記モデレータ・テンプレートのいずれが現在実行権限を有しているかを判断するプラン・イタレータとを含む第 1 のプレースから第 2 のプレースへ移動し、前記第 1 のプレースにおいて第 1 のアクティビティを実行し、前記第 2 のプレースにおいて第 2 のアクティビティを実行する移動オブジェクト群が提供される。

本発明の他の一態様においては、

(a) 前記第 1 のアクティビティにおいて、前記モデレータ・テン

プレートとのリンク情報を管理するサクセッサ・リストを生成する段階と、

5 (b) 前記モデレータ・テンプレートにおいて、前記第1のアクティビティとのリンク情報を管理するプレデセッサ・リストと、前記第2のアクティビティとのリンク情報を管理するサクセッサ・リストとを生成する段階と、

10 (c) 前記第2のアクティビティにおいて、前記モデレータ・テンプレートとのリンク情報を管理するプレデセッサ・リストを生成する段階とを含む第1のプレースにおいて実行される第1のアクティビティと、第2のプレースにおいて実行される第2のアクティビティと、前記第1のプレースにおいて移動オブジェクト群の移動を制御する移動モデレータ・エージェントに前記第2のプレースへ移動することを要求するモデレータ・テンプレートを含む移動オブジェクト群を生成する方法が提供される。

15

本発明の他の一態様においては、

(a) 前記第1のアクティビティと、前記モデレータ・テンプレートと、前記第2のアクティビティと、の実行順序を管理する順序リストを生成する段階と、

20

(b) 前記第1のアクティビティにおいて、前記モデレータ・テンプレートとのリンク情報を管理するサクセッサ・リストを生成する段階と、

25 (c) 前記モデレータ・テンプレートにおいて、前記第1のアクティビティとのリンク情報を管理するプレデセッサ・リストと、前記第2のアクティビティとのリンク情報を管理するサクセッサ・リストとを生成する段階と、

(d) 前記第 2 のアクティビティにおいて、前記モデレータ・テンプレートとのリンク情報を管理するブレデセッサ・リストを生成する段階とを含む第 1 のプレースにおいて実行される第 1 のアクティビティと、第 2 のプレースにおいて実行される第 2 のアクティビティと、前記第 1 のプレースにおいて移動オブジェクト群の移動を制御する移動モデレータ・エージェントに前記第 2 のプレースへ移動することを要求するモデレータ・テンプレートを含む移動オブジェクト群を生成する方法が提供される。

10 本発明の他の一態様においては、

(a) 前記第 1 のアクティビティにおいて、前記モデレータ・テンプレートとのリンク情報を管理するサクセッサ・リストを生成する手段と、

15 (b) 前記モデレータ・テンプレートにおいて、前記第 1 のアクティビティとのリンク情報を管理するブレデセッサ・リストと、前記第 2 のアクティビティとのリンク情報を管理するサクセッサ・リストとを生成する手段と、

20 (c) 前記第 2 のアクティビティにおいて、前記モデレータ・テンプレートとのリンク情報を管理するブレデセッサ・リストを生成する手段とを含む第 1 のプレースにおいて実行される第 1 のアクティビティと、第 2 のプレースにおいて実行される第 2 のアクティビティと、前記第 1 のプレースにおいて移動オブジェクト群の移動を制御する移動モデレータ・エージェントに前記第 2 のプレースへ移動することを要求するモデレータ・テンプレートを含む移動オブジェクト群を生成する装置が提供
25 される。

本発明の他の一態様においては、

(a) 前記第1のアクティビティにおいて、前記モデレータ・テンプレートとのリンク情報を管理するサクセッサ・リストを生成することを前記コンピュータに指示するプログラムコード手段と、

5 (b) 前記モデレータ・テンプレートにおいて、前記第1のアクティビティとのリンク情報を管理するプレデセッサ・リストと、前記第2のアクティビティとのリンク情報を管理するサクセッサ・リストとを生成することを前記コンピュータに指示するプログラムコード手段と、

10 (c) 前記第2のアクティビティにおいて、前記モデレータ・テンプレートとのリンク情報を管理するプレデセッサ・リストを生成することを前記コンピュータに指示するプログラムコード手段とを含む第1のプレースにおいて実行される第1のアクティビティと、第2のプレースにおいて実行される第2のアクティビティと、前記第1のプレースにおいて移動オブジェクト群の移動を制御する移動モデレータ・エージェントに前記第2のプレースへ移動することを要求するモデレータ・テンプレートを含む移動オブジェクト群を生成するプログラムを格納するコン
15 ピュータによって読み取り可能な記憶媒体が提供される。

[図面の簡単な説明]

20

第1図は、本発明の移動エージェントが動作する分散ネットワーク環境を示す図である。

第2図は、本発明の移動エージェントが分散ネットワーク上で移動する態様の一例を示す図である。

25

第3図は、本発明におけるクライアントシステムまたはサーバシステムのハードウェア構成の一実施例を示すブロック図である。

第4図は、本発明のクライアントサイトにおける処理要素の一実施例を示すブロック図である。

第5図は、本発明のサーバサイトにおける処理要素の一実施例を示すブロック図である。

5 第6図は、本発明の好適な実施例におけるオブジェクト図である。

第7図は、本発明の好適な実施例におけるオブジェクト図である。

第8図は、本発明の好適な実施例におけるオブジェクト間のメッセージの流れを示す図である。

10 第9図は、本発明の好適な実施例における開発支援ツールのユーザインターフェースを示す図である。

第10図は、本発明の好適な実施例における開発支援ツールのユーザインターフェースを示す図である。

第11図は、本発明の好適な実施例における開発支援ツールのユーザインターフェースを示す図である。

15 第12図は、本発明の好適な実施例における移動エージェントにイベントが渡り実行される経過を示す図である。

第13図は、本発明の好適な実施例におけるプラン・オブジェクトにプランノードを追加する手順を示すフローチャートである。

20 第14図は、プラン・オブジェクトとプランノード・オブジェクトのリンク状態等の一例を示す概念図である。

第15図は、プラン・オブジェクトとプランノード・オブジェクトのリンク状態等の一例を示す概念図である。

第16図は、プラン・オブジェクトとプランノード・オブジェクトのリンク状態等の1例を示す概念図である。

25 第17図は、本発明の好適な実施例におけるプラン・オブジェクトの実行手順を示すフローチャートである。

第18図は、本発明の好適な実施例におけるモデレータ・テンプレートの機能ブロック図である。

第19図は、本発明の好適な実施例におけるプレースの概念図である。

5 第20図は、本発明の好適な実施例におけるモデレータ・テンプレートの処理概要を示すフローチャートである。

第21図は、本発明の好適な実施例における移動オブジェクト群がオリジンのプレースからディスティネーションのプレースに移動する状態を示す概念図である。

10 第22図は、本発明の好適な実施例における移動オブジェクト群がオリジンのプレースからディスティネーションのプレースに移動する状態を示す概念図である。

第23図は、本発明の好適な実施例におけるIterationテンプレートのオリジンにおける処理手順を示すフローチャートである。

15 第24図は、本発明の好適な実施例における移動オブジェクト群がオリジンのプレースからディスティネーションのプレースに移動する状態を示す概念図である。

第25図は、本発明の好適な実施例における移動オブジェクト群がオリジンのプレースからディスティネーションのプレースに移動する状態を示す概念図である。

20 第26図は、本発明の好適な実施例におけるAND Splitテンプレートのオリジンにおける処理手順を示すフローチャートである。

第27図は、本発明の好適な実施例におけるカレント・ノードを決定するロジックを示すフローチャートである。

25 第28図は、本発明の好適な実施例におけるAND Joinテンプレートのディスティネーションにおける処理手順を示すフローチャートで

ある。

第29図は、本発明の好適な実施例におけるAND Joinテンプレートのディスティネーションにおける処理手順を示すフローチャートである。

5 第30図は、本発明の好適な実施例におけるAND Joinテンプレートのディスティネーションにおける処理の概要を説明する図である。

第31図は、本発明の好適な実施例における移動オブジェクト群がオリジンのプレースからディスティネーションのプレースに移動する状態を示す概念図である。

10 第32図は、本発明の好適な実施例におけるOR Splitテンプレートのオリジンにおける処理手順を示すフローチャートである。

第33図は、本発明の好適な実施例における移動オブジェクト群がオリジンのプレースからディスティネーションのプレースに移動する状態を示す概念図である。

15 第34図は、本発明の好適な実施例におけるOR Joinテンプレートのディスティネーションにおける処理手順を示すフローチャートである。

[符号の説明]

20

100…ノード・システム、211…プランナ・エージェント

221…プラン・オブジェクト、

251…モデレータ・エージェント

261…プラン・オブジェクト、270…移動オブジェクト群

25 271…モデレータ・テンプレート、281…アクティビティ

291…アクター・エージェント

[発明を実施するための最良の形態]

5 以下、図面を参照して本発明の実施例を説明する。

A. システムの概要

10 図1を参照すると、本発明の移動オブジェクトを実行する分散ネットワーク環境150が示されている。各サーバ112～117には、移動エージェント125、135、141等にサービスを与えることが可能なブレース102～107が用意されている。この分散ネットワーク環境におけるブレース群はクラウドと呼ばれている。

15 クライアント・システム101においては、移動エージェントを生成するためのプランナ・エージェント113が存在する。プランナ・エージェント113は、モデレータ・エージェント111に、生成したプランを渡し、モデレータ・エージェントはそのプランを実行することにより、指定されたブレースを移動していくことができる。

20 移動エージェント125等は各ブレース102～107において存在する他のエージェントとコンタクトし（移動エージェントとコンタクトし、サービスを提供するエージェントを特にアクター・エージェントと呼ぶ）、リクエストを送出し、そのリクエストの結果を受けることができる。ブレースは、エージェント間のコンタクトをサポートしている。また、移動エージェント125等は、アクター・エージェントから受領したリクエストの結果をResult139、143、153、163
25 として保持し、移動を継続することができ、移動途中で、このResult

1 t を合成、分割等さまざまな加工を施すことも可能である。

本発明において、移動エージェントの移動、分裂、消滅は、モデレータ・テンプレート 181～187 によって制御されており、ユーザは、その処理の態様により、さまざまなテンプレートを組合せ、図 2 に示す
5 ような複雑な作業を簡単に定義することができる。

図 3 は、図 1 に示された分散ネットワーク環境上に存在する各ノード（サーバまたはクライアントサイト）のハードウェア構成の概観図が示されている。各ノード 100 は、中央処理装置（CPU）1 とメモリ 4 とを含んでいる。CPU 1 とメモリ 4 は、バス 2 を介して、補助記憶装置としてのハードディスク装置 13 とを接続してある。フロッピーディスク装置（または MO、CD-ROM 等の記憶媒体駆動装置）20 はフ
10 ロッピーディスクコントローラ 19 を介してバス 2 へ接続されている。

フロッピーディスク装置（または MO、CD-ROM 等の記憶媒体駆動装置）20 には、フロッピーディスク（または MO、CD-ROM 等の記憶媒体）が挿入され、このフロッピーディスク等やハードディスク装置 13、ROM 14 には、オペレーティングシステムと協働して CPU 等に命令を与え、本発明を実施するためのコンピュータ・ソフトウェアを記録することができ、メモリ 4 にロードされることによって実行される。このコンピュータ・ソフトウェアは圧縮し、または、複数に分割
15 して、複数の媒体に記録することもできる。

ノード・システム 100 は更に、ユーザ・インターフェース・ハードウェアを備えたシステムとすることができ、ユーザ・インターフェース・ハードウェアとしては、例えば、入力をするためのポインティング・デバイス（マウス、ジョイスティック等）7 またはキーボード 6 や、視覚
20 データをユーザに提示するためのディスプレイ 12 があり、また、パラレルポート 16 を介してプリンタを接続するも可能である。このノード・

システム１００は、シリアルポート１５を介してモデムを接続することが可能であり、シリアルポート１５およびモデムまたはトークンリングや通信アダプタ１８等を介してネットワークに接続し、他のコンピュータシステムと通信を行っている。

- 5 このように、本発明は、通常のパーソナルコンピュータ（ＰＣ）、ワークステーション、汎用機や、これらの組合せによって実施可能であることを容易に理解できるであろう。ただし、これらの構成要素は例示であり、その全ての構成要素が本発明の必須の構成要素となるわけではない。特にサーバ側においては、サーバを遠隔操作することも可能であるため、ユーザインタフェース等は不要であり、ＣＰＵやメモリ等の基本的なデータ処理機能と通信機能があれば足りる。
- 10 クライアントシステム側、サーバシステム側ともに、オペレーティング・システムとしては、Windows NT（マイクロソフトの商標）、Windows 95（マイクロソフトの商標）、Windows 3. x（マイクロソフトの商標）、OS/2（IBMの商標）、AIX（IBMの商標）上のX-WINDOWシステム（MITの商標）などの、標準でGUIマルチウインドウ環境をサポートするものや、PC-DOS（IBMの商標）、MS-DOS（マイクロソフトの商標）などのキャラクタ・ベース環境のもの、さらにはOS/Open（IBMの商標）VxWorks（Wind River Systems, Inc. の商標）等のリアルタイムOSでも実現可能であり、特定のオペレーティング・システム環境に限定されるものではない。
- 15
- 20

B. システム構成

25

次に、図４、５のブロック図を参照して、本発明の好適な実施例にお

けるクライアント側及びサーバ側のシステム構成について説明する。

図4は、クライアントシステム210における処理要素の一実施例を示す機能ブロック図である。プランナ・エージェント211は、プラン定義部213と、プランID生成部215を含んでいる。プランID生成部215は、プランにネットワーク上ユニークなプランIDを割りふる。本発明の好適な実施例においてプランID生成部215は、インターネットのURL、ポート番号、エージェントID（プランナ・エージェントに割りふられている）、シリアルNo.（プランナ・エージェントがプランID作成毎割りふる）を合成してプランIDを作成している。

プラン定義部213は、プラン・ノード・ライブラリ201からモデレータ・テンプレート203やアクティビティ205を取得し、モデレータ・エージェントや、プランを生成する。作成されたプランには、実行するモデレータ・テンプレートやアクティビティの情報等を管理するプラン構成部223と、モデレータ・テンプレート、アクティビティの順序を定義する順序リスト225を含んでいる。プラン構成部223は、プランノード・オブジェクト（モデレータ・テンプレート、アクティビティの総称）の関係付けの情報を有しており、ネットワークトポロジー231を形成している。

図5は、サーバ・システム250における処理要素の一実施例を示す機能ブロック図である。移動オブジェクト群270はモデレータ・エージェント251、プラン・オブジェクト261、モデレータ・テンプレート271、アクティビティ281を含んでいる。モデレータ・エージェント251は、移動エージェント・ライフサイクル制御部253とプラン実行部255を含んでいる。移動エージェント・ライフサイクル制御部253は、移動オブジェクト群270の移動、複製、消去等を実行

する。プラン実行部 255 は、新たなプレースに到着したことを認識し、プラン・オブジェクト 261 にプランの実行を命令する。

この一方、プラン・オブジェクトは、プラン実行制御部 263、カレント・ノード 267、順序リスト 265 を含んでいる。プラン実行制御部 263 は、モデレータ・エージェントのプラン実行部からプランの実行のインストラクションを受領すると、カレント・ノード 267、順序リスト 265 を参照し、現在実行すべきプランノード・オブジェクト 271、281 を決定し、プランノード・オブジェクト 271、281 に実行を要求する。

カレント・ノード 267 は、後述するプランノードイタレータに対応するものであり、現在どのプランノード・オブジェクトを実行しているかを管理している。順序リスト 265 は、実行すべきプランノード・オブジェクト 271、281 を特定するリストである。分岐等が発生しない場合、プランノード・オブジェクト 271、281 は、この順序リスト 265 の順番にしたがってプランノードに実行を要求していく。

モデレータ・テンプレート 271 は、モデレータ・エージェント制御部 273、プランステップ実行部 275、アドレス参照部 277 を含んでいる。モデレータ・エージェント制御部 273 は、モデレータ・エージェント 251 の移動エージェント・ライフサイクル制御部 253 に対し、移動オブジェクト群 270 の移動、複製、消去を要求する。プラン・ステップ実行部 275 は、モデレータ・テンプレートの種類によって異なる定義となっており、移動前、移動後の夫々のプレースにおいて実行すべき命令等を管理している。アドレス参照部 277 は、アクティビティからアドレスを取得し、モデレータ・エージェント制御部 27 に移動すべきアドレスを教える等の役割を担っている。

アクティビティ 281 は、アクティビティ実行部 283 とメッセージ

機能 285 を含んでいる。アクティビティ実行部 283 は、移動エージェントが渡り歩く各プレースで実行するインストラクションや各プレースで受け取った Result を格納する機能を有している。メッセージ機能 285 は、アクター・エージェント 291 とのメッセージをやり取りする機能を有している。

アクター・エージェント 291 は、プレース上に存在する他のエージェントであり、アクティビティのリクエストに対し、サービスを提供する機能を有している。このアクター・エージェント 291 も、アクティビティとメッセージのやり取りするメッセージ機能 293 を有し、また、リクエストの内容を判断し、サービスを提供するメッセージ処理部 295 を有している。

以上図 4、5 に示す各機能ブロックを説明したが、これらの機能ブロックは、論理的機能ブロックであり、各々 1 つのまとまりを持ったハードウェアやソフトウェアによって実現されることを意味するのではなく、複合し、または共通したハードウェアやソフトウェアによって実現可能である。また、この図 4、5 に示す全ての機能ブロックが本発明に必須の構成要素という訳ではない。例えば、順序付リスト 265 は、カレント・ノード管理部 267 が直接カレント・ノードを管理し、また、AND Join 等の処理においても、他の Result を取得するモデレータ・エージェントを決めておけばよいので、特に必要な構成要素ではない。

C. オブジェクト構成

図 6、7 は、本発明の移動オブジェクトのオブジェクト図である。モデレータ・エージェント 303、プラン 305、プラン ID、プラン・

イタレータ 307 の各クラスは、メソッドを有しており、プランノード 311、アクティビティ 313、モデレータ・テンプレート 315、及び Chain 等のテンプレート 321～326 の各クラスは、データ及びメソッドを有している。

- 5 この図において、三角印 302、314、316 は、上位のクラスのデータやメソッドを承継していることを示し、黒丸 312 はその下位のクラスが複数存在しえることを示している。プランノードは抽象的なクラスで、モデレータ・テンプレートとアクティビティクラスの双方を共通のインターフェースで定義する。これによってプランは、プランの生成や実行中に双方のクラスを同等に扱うことができる。

- 10 移動エージェント 301 は分散コンピュータ環境上の各プレースで現在提供されている機能（プレース・クラス）であり、このクラスに移動エージェントの作成を指示する API を投げることによって、ユーザは簡単に移動エージェント（モデレータ・エージェント）を作成することができる。現在、従来技術の説明で上述した Java ベースのモバイル・エージェントやテレスクリプトの移動エージェントの双方でサポートされている機能である。

- 15 プランノードクラスはサクセッサとプレデセッサのリストを有しており、複数のプランノードの関係を管理している。プランノードのクラスは "setPredecessor" "setSuccessor" のメソッドを提供し、ノード間の関係を定義している。プランノードの具体的なクラスは、これらのメソッド間の関係を見分けることができる。

- 20 モデレータ・テンプレートは、異なるプレースで実行されるアクティビティと連結するため、オリジン（移動前のプレース）かディスティネーション（移動後のプレース）かを区別する状態情報を有している。
- 25 本発明の好適な実施例においては、モデレータ・テンプレートはアクテ

ィビティ・オブジェクトとだけ結合することができる。モデレータ・テンプレートにはChainを始め、複数のテンプレートが提供されており、複数の分裂したモデレータ・エージェントが分担して仕事をするための基本的メカニズム等の提供を可能にしている。

5 図8は、本発明の好適な実施例における各クラスの主要なメッセージフローを示す図である。モデレータ・エージェント351はプラン353に対しnextStep（次のステップを実行することを指示するインストラクション）を送出し、プランはこれに応答して、プランノード・オブジェクト（モデレータ・テンプレート357またはアクティビティ355）に対しnextStepを送出する。アクティビティ355は、
10 ユーザの指定したリクエストを実行するため、アクター・エージェント359に対しsendMessage（メッセージの送付要求）を送出する。モデレータ・テンプレート357は、その種類や状況に応じて、アクティビティ355にgetARL（アドレスの送出命令）を送出や、
15 isMarked（マーク付けをする）をお送出する。また、モデレータ・エージェント351に対しclone（複製の作成要求）、dispatch（移動要求）、dispose（消滅要求）を送出する。

D. プランナオブジェクトの作成

20 プランナオブジェクトは、クライアントサイトで実行されるインストラクション群であり、このインストラクションが実行されると、モデレータ・エージェントの生成要求が送出され、また、実行される一連のアクティビティを定義するプラン・オブジェクトを生成し、モデレータ・
25 エージェントに渡す。

 モデレータ・エージェントは、上述のように、各ブレースで現在提供

されている機能（プレース・クラス）を利用して生成される。本発明の好適な実施例においては、J a v a ベースのモバイル・エージェントの A g l e t s クラスに移動エージェントの作成を指示する A P I を投げることによって、モデレータ・エージェントの生成を行っている。

- 5 プランの生成においては、大きく分けて、プラン I D の生成と、プラン・オブジェクトの下位に位置するオブジェクトとのリンク状態を確立する工程に分けられる。

10 本発明の好適な実施例においては、プラン・オブジェクトにグローバルなネットワーク上で一意に決まるユニークなプラン I D を割りふられる。本発明の好適な実施例においてプラン I D 生成部 2 1 5（図 4）は、インターネットの U R L、ポート番号、エージェント I D（プランナ・エージェントに割りふられている）、シリアル N o.（プランナ・エージェントがプラン I D 作成毎割りふる）を合成してプラン I D を作成している。

- 15 このプラン I D 生成部のコーディングの例を表 1 に示す。本発明の好適な実施例においては、プラン I D 生成部のインストラクション群はプログラム部品としてユーザに提供されている。

[表 1]

20

```
public class Planner extends StationaryAgent{  
    private Plan _plan;  
    private PlannerFrame f;  
    private URL _home;  
25    private AgletIdentifier _id;  
    private int _planIndex = 0;
```

```
public void onCreate(Object args){  
    //プランの選択のためウインドウを生成する。  
    f = new PlannerFrame("JMT - PlannerAgent", this);  
5    f.show();  
    try{  
        //プランナ・エージェントの存在しているURLを入手  
        _home = getAgletContext().getHostingURL();  
        //プランナ・エージェントのIDを入手  
10        _id = getIdentifier();  
    }catch(InvalidAgletException e){  
        Utility.print(this, e.getMessage());  
    }  
}
```

15 次に、プラン定義部 213（図4）のChainテンプレートを使用する場合のコーディングを表2に示す。本発明の好適な実施例の一態様においては、テンプレートの種類ごとサンプルコードがユーザ（プログラマ）に提供されており、ユーザは、そのサンプルコードを利用して所望のコーディングに作り替えることにより、このインストラクション群
20 を形成する。

[表2]

```
public void doPlan1(){  
25 //プラン・オブジェクトの生成  
    Plan plan = new Plan(generatePlanID());
```



```
//アクティビティの生成
    Activity act1, act2;

    act1 = new Activity("atp://kazu.yamato.ibm.com:434/", "JMT.samples.TextFieldAgent", new Message(Activity.REQUEST));

5    act2 = new Activity("atp://" + _home.getHost() + ":434/", "JMT.samples.ReportAgent", new Message(Activity.REPORT));

    //プランにアクティビティとテンプレートを追加
    try{
        Vector predecessorIndexList;

10        predecessorIndexList = new Vector();

        //プランにChainテンプレートを追加
        Integer hwd1 = plan.addNode(null, new Chain());

        //プランにact1を追加
        predecessorIndexList.addElement(hwd1);

15        Integer hwd2 = plan.addNode(predecessorIndexList, act1);

        //プランにChainテンプレートを追加
        predecessorIndexList.removeAllElements();
        predecessorIndexList.addElement(hwd2);
        Integer hwd3 = plan.addNode(predecessorIndexList, new Chain());

20    n());

        //プランにact2を追加
        predecessorIndexList.removeAllElements();
        predecessorIndexList.addElement(hwd3);
        Integer hwd4 = plan.addNode(predecessorIndexList, act2);

25    }catch(BlockedByIteratorException e){
        Utility.print(this, e.getMessage());
    }
```

```
    }  
    //モデレータ・エージェントを生成  
    createModeratorAgent(plan);  
}
```

5 本発明の好適な実施例の他の態様においては、このコーディングをユーザが簡単に行うことを可能にする開発支援ツールを提供している。図9乃至図11は本発明の好適な実施例における開発支援ツールのユーザインターフェースを示す図である。ユーザはこの開発支援ツールを使用することによって、上記プラン定義部のコーディングを簡単に生成する
10 ことができる。

かかる開発支援ツールは、例えば、Workflow Automation (「Workflow Automation」は、Ultimus LLC社の商標) 等によって提供されている公知の手法であるため (<http://www.ultimus1.com>等で情報入手可能)、その具体的な機能実現手段の記載は省略し、ユーザの行う手順
15 を説明することとする。

まず、ユーザがプルダウン・メニューから作成371を選択し、ポインティング・デバイスでクリックすると、プランナ・エージェント開発支援ツールはプラン名を入力するためのダイアログボックスを表示する。
20 これにより、ユーザはプランIDとは異なるユーザにとって親しみやすい名前
でプランを管理することができる。次に、ユーザがプラン名をプラン名
の入力エントリ374入力し、OKボタン373を押すと、プラン定義用のウインドウ380
が開かれる。

そして、ユーザがポインティング・デバイスでプランノード・ボックス390
25 のアクティビティのプランのアイコン399をプラン定義用ウインドウ380にドラッグ
しドロップする。すると、図10に示すよう

にアクティビティのプロパティ・エディタ・ウインドウ 381 が開かれる。

5 アクティビティ・プロパティ・エディタ・ウインドウ 381 にはアクティビティ名 382、アドレス名（アクティビティを実行するプレースのアドレス） 383、アクター・エージェント名 384、メッセージ 385 のエントリが用意されている。ユーザはこの各エントリに所望の入力を行い、OK ボタン 386 をクリックすることによってアクティビティの定義をすることができる。

10 これと同様に、プランノード・ボックス 390 から Chain のアイコン 391 をプラン定義用ウインドウ 380 の act 1 の右側又は下側にドラッグしドロップする。テンプレートのアイコンをドロップする場合、本発明の好適な実施例においては、OR Split 以外は、テンプレート用プロパティ・エディタ・ウインドウは表示されない。OR Split のアイコンがドロップされた場合には、テンプレート用プロパティ・エディタ・ウインドウが表示され、アクティビティの選択条件を入力することとなる。次に、act 1 のときと同様に act 2 アクティビティを Chain の右側又は下側にドラッグ&ドロップし、プロパティを設定する。

20 そして、プラン定義用ウインドウ 380 上のリンクを張りたい 2 つのオブジェクト（例えば act 1 : 375 と Chain 376）をクリックし選択状態にし、編集メニューのリンク 379 を選択すると、両者にリンクが張られる（サクセッサ・リストとプレデセッサ・リストが生成される）。

25 ユーザは、全てのオブジェクトのリンクを張る作業を終了した後、プラン定義用ウインドウ 380 の構成 397 をクリックする。プランナ・エージェント開発支援ツールは、プラン・オブジェクトに各要素を追加

し、プランナオブジェクトを生成する。

そして、ユーザは、プラン定義用ウインドウ 380 の実行 398 をクリックすると、生成されたプランナオブジェクトおよびクラスにイベントが渡り、モデレータ・エージェントが生成され、プランナ・エージェントがプランの生成を開始し、生成したプランをモデレータ・エージェントに渡す。モデレータ・エージェントはこの生成されたプランを自動的に実行していく。

この手順を図 12 に示す。プラン定義用ウインドウ 380 で発生したイベントはプランナオブジェクト 470 に渡され、プランナオブジェクトはイベントの解析を行い 471、表 1、2 に対応するオブジェクト・コードが実行され、プランの定義及び構成を開始する 472、473。これにより、生成されたプラン 474 をモデレータ・エージェント 351 に渡すと、モデレータ・エージェントはプランを 1 ステップずつ実行していく。

本発明の好適な実施例においては、プラン・オブジェクトを作成した後、プランクラスの "addNode" というメソッドを使用して複数のプランノードを追加することもできる。また、プラン・オブジェクトを作成した後、プランクラスの "RemoveNode" というメソッドを使用して複数のプランノードを取り除き、サクセッサ・リストを更新することもできる。

E. プランの生成

E-1. Chain テンプレートを含むトポロジーの場合

次に、表 1、2 に示したプランナ・エージェントがプラン・オブジェクトを生成していく手順を説明する。ここでは、図 9 乃至図 11 で説明した開発支援ツールによって生成された Chain テンプレートを含む

プラン・オブジェクトを例に説明を行う。図13は、プラン・オブジェクトにプランノードを追加する手順を示すフローチャートであり、図14は、プラン・オブジェクトとプランノード・オブジェクトのリンク状態等を示す概念図である。

- 5 このChainテンプレートは、最もシンプルなテンプレートで、異なるプレースにおける2つのアクティビティを結び付けるものである。オリジンのプレースにおいて、ChainはサクセッサのアクティビティのARL（アドレス）を受領し、目的のプレースに移動オブジェクト群を移動させる。このオリジンのプレースとディスティネーションのプレースは同一のプレースでもよい。

- 10 プランナ・エージェントが実行されると、まず、プラン・オブジェクトを生成し、プランIDをセットする。次に、プランを構成するアクティビティ・オブジェクト520、540を生成する。図14に示すように、各アクティビティには、サービスを依頼する相手のエージェント名521、実行するサーバのアドレス523、依頼内容524を格納する。
- 15 図14の例では2つのアクティビティ・オブジェクトを作成している。

- 20 その後、プラン・オブジェクト510に2つのアクティビティ・オブジェクトを格納する。2つのアクティビティの連結には後述するChainテンプレート530が用いられる。ただし、プラン・オブジェクト510に格納する際に必ずどの要素の後に連結されるかを指定する必要がある、そのためにプレデセッサ・インデックス・リスト(Predesecor Index List)515が用いられる。本発明の好適な実施例においては、このプレデセッサ・インデックス・リスト515はリスト構造体として
- 25 実施されている。

プランナオブジェクトは、プラン・オブジェクト510にプレデセッ

5 サ・インデックス・リスト 5 1 5 を生成した後、最初のアクティビティ・オブジェクト (a c t 1) を追加する。図 1 3 に、プラン・オブジェクトにプランノードを追加する手順を示す。プランナオブジェクトはプレデセッサ・インデックス・リスト 5 1 5 を検査し、プレデセッサ・インデックス・リストが空か否かを判断する (ブロック 8 0 3)。a c t 1 の場合、最初の要素であり、プレデセッサ・インデックス・リスト 5 1 5 は空であるので、a c t 1 は、プランノード・リスト 5 1 3 に追加する (ブロック 8 1 5)。そして、プラン・オブジェクト 5 1 0 はプランナオブジェクトに対しプランノードのプランノード・リスト内のインデックスを戻り値として返し (ブロック 8 1 7)、処理を戻す 8 1 9。この場合、インデックスが戻り値として " 0 " が返される。

15 次にプラン・オブジェクト 5 1 0 に C h a i n オブジェクト 5 3 0 を追加する。この場合再び図 1 3 の処理が再開する。プランナオブジェクトはプレデセッサ・インデックス・リスト 5 1 5 を検査し、プレデセッサ・インデックス・リスト 5 1 5 が空か否かを判断する (ブロック 8 0 3)。この場合、すでに a c t 1 が登録されているので、処理はブロック 8 0 5 に進む。ブロック 8 0 5、8 0 7 においては、プレデセッサ・リストの要素数を処理が終了したか否か判断している。

20 次にプラン・オブジェクト 5 1 0 に対し、C h a i n オブジェクト 5 3 0 を追加する (ブロック 8 0 9)。まず、a c t 1 (5 2 0) の後に連結することを指定するため、プレデセッサ・インデックス・リスト 5 1 5 に先程の a c t 1 を追加したときの戻り値 " 0 " を入れる。

predecessor index list [0]

25 次に、C h a i n オブジェクトを a c t 1 の後に追加する。すると、

プラン・オブジェクト 510 は、プレデセッサ・リスト 515 の中で指定された `act 1` のサクセッサ・リスト 529 に `Chain` オブジェクトを格納する（ブロック 809）。こんどは逆に `Chain` のプレデセッサ・リスト 531 に `act 1` を格納する（ブロック 811）。最後に
5 プラン・オブジェクト 510 のプランノード・リスト 513 に `Chain` 523 が格納され、そのインデックスが戻り値（この場合は"1"）としてプランナ・エージェントに返される。

次に `act 2` (540) を追加する。まず、プレデセッサ・インデックス・リスト 515 を空にし（`act 1` の情報をクリアし）、`Chain`
10 `n` をプランに追加したときの戻り値（1）をセットする。

`predecessor index list [1]`

次に、`act 2` を `Chain` の後に追加する。`Chain` の場合と同様に、プラン・オブジェクト 510 は、プレデセッサ・リスト 515 の
15 中で指定された `Chain` のサクセッサ・リスト 529 に `act 2` オブジェクトを格納する（ブロック 809）。こんどは逆に `act 2` のプレデセッサ・リスト 531 に `Chain` を格納する（ブロック 811）。最後にプラン・オブジェクト 510 のプランノード・リスト 513 に
20 `act 2` が格納され、そのインデックスが戻り値（この場合は"2"）としてプランナ・エージェントに返される。

E-2. `Iteration` テンプレートを含むトポロジーの場合

25 図 15 は、プラン・オブジェクトとプランノード・オブジェクトのリンク状態等を示す概念図である。このオブジェクト群も `Chain` を含

むトポロジーの場合と同様にリンク付けがなされる。ただし、Iterationのテンプレートの場合、図15に示すように、ループの戻り先の情報として、リターンインデックス575を有している。本発明の好適な実施例においては、リターンインデックス575に戻り先のオブジェクト(Chain550)に対応したプラン・オブジェクト510のプレデセッサ・インデックス・リスト515の値をセットしている。表3にIterationテンプレートを含むトポロジーの場合のプランナ・エージェントのコーディング例を示す。

10 [表3]

```
public void doPlan2(){
//プランの生成
    Plan plan = new Plan(generatePlanID());
15 //アクティビティの生成
    Activity act1, act2;
    act1 = new Activity("atp://kazu.yamato.ibm.com:434", "JMT.samples.TextFieldAgent", new Message(Activity.REQUEST));
    act2 = new Activity("atp://"+_home.getHost()+":434/", "JMT.samples.ReportAgent", new Message(Activity.REPORT));
20 //プランにアクティビティとテンプレートを追加
    try{
        Vector predecessorIndexList;
        predecessorIndexList = new Vector();
25
        Integer hwd1 = plan.addNode(null, new Chain());
```



```
predecessorIndexList.removeAllElements();
predecessorIndexList.addElement(hwd1);
Integer hwd2 = plan.addNode(predecessorIndexList, act1);
predecessorIndexList.removeAllElements();
5    predecessorIndexList.addElement(hwd2);
    Integer hwd3 = plan.addNode(predecessorIndexList, new Cha
in());
    predecessorIndexList.removeAllElements();
    predecessorIndexList.addElement(hwd3);
10    Integer hwd4 = plan.addNode(predecessorIndexList, act2);
    // Create a StringLengthIteration template by specifying th
e return node.

    //
15
    StringLengthIteration iteration = new StringLengthIterati
on(hwd1);
    predecessorIndexList.removeAllElements();
    predecessorIndexList.addElement(hwd4);
20    Integer hwd5 = plan.addNode(predecessorIndexList, iterati
on);
    }catch(BlockedByIteratorException e){
        Utility.print(this, e.getMessage());
    }
25 //モデレータ・エージェントを生成
    createModeratorAgent(plan);
```

}

E-3. ANDSplit、ANDJoinテンプレートを含むトポロジーの場合

5

図16は、プラン・オブジェクトとプランノード・オブジェクトのリンク状態等を示す概念図である。このオブジェクト群もChainを含むトポロジーの場合と同様にリンク付けがなされる。ただし、ANDSplitテンプレートは、1つのプレデセッサのアクティビティと複数のサクセッサのアクティビティを持つ。この複数のサクセッサのアクティビティは、このANDSplitテンプレートによって、異なるプレースにおいて並列に実行されることができる。

10

この一方、ANDJoinテンプレートは複数のプレデセッサのアクティビティと一つのサクセッサのアクティビティを有している。デステーション・プレースにおいて、このANDJoinテンプレートの制御のもと、複数のモデレータ・エージェントが一つのエージェントに合体する。表4にANDSplitテンプレート及びANDJoinテンプレートを含むトポロジーの場合のプランナ・エージェントのコーディング例を示す。

15

20

[表4]

```
public void doPlan3(){
```

```
//プランを生成
```

```
Plan plan = new Plan(generatePlanID());
```

25

```
//アクティビティを生成
```

```
Activity act1 = new Activity("atp://kazu.yamato.ibm.com:434/"
```

```
, "JMT.samples.TextFieldAgent", new Message(Activity.REQUEST));
    Activity act2 = new Activity("atp://" + _home.getHost() + ":600/"
, "JMT.samples.TextFieldAgent", new Message(Activity.REQUEST));
    Activity act3 = new Activity("atp://" + _home.getHost() + ":434/"
5 , "JMT.samples.ReportAgent", new Message(Activity.REPORT));

//プランにアクティビティとテンプレートを追加
    try{
        Vector predecessorIndexList;
10     predecessorIndexList = new Vector();
        //プランにANDSplitを追加
        Integer hwd1 = plan.addNode(null, new ANDSplit());
        predecessorIndexList.addElement(hwd1);
        //ANDSplitのサクセッサとして3つのアクティビティを追加
15     Integer hwd2 = plan.addNode(predecessorIndexList, act1);
        Integer hwd3 = plan.addNode(predecessorIndexList, act2);
        //ANDJoinのプレデセッサ・リストに3つのアクティビティのインデ
        ックスを追加
        predecessorIndexList.removeAllElements();
20     predecessorIndexList.addElement(hwd2);
        predecessorIndexList.addElement(hwd3);
        //プランにANDJoinを追加
        Integer hwd4 = plan.addNode(predecessorIndexList, new AND
Join());
25     predecessorIndexList.removeAllElements();
        predecessorIndexList.addElement(hwd4);
```

```
//プランにANDJoinのサクセッサとしてアクティビティを追加
Integer hwd5 = plan.addNode(predecessorIndexList, act3);
} catch(BlockedByIteratorException e) {
    Utility.print(this, e.getMessage());
5      }
//モデレータ・エージェントを生成し、プランを渡す
createModeratorAgent(plan);
}
```

10 E-4. ORJoinテンプレートを含むトポロジーの場合

ORJoinテンプレートは、ANDJoinテンプレートと同様に、
複数のプレデセッサのアクティビティと1つのサクセッサのアクティビ
ティを持つ。ただし、ORJoinテンプレートの場合は、複数のプレ
15 デセッサのアクティビティから1つのプレデセッサのアクティビティを
選択し、他のプレデセッサのアクティビティを無視する。例えば、国内
線の航空会社の数社に羽田・札幌間の航空チケットを予約状況を検索し、
最も早く予約状況の検索ができたものを採用するような処理に使用する。
表5にORJoinテンプレートを含むトポロジーの場合のプランナ・
20 エージェントのコーディング例を示す。

[表5]

```
public void doPlan4() {
25 //プランを生成
    Plan plan = new Plan(generatePlanID());
```

```
//アクティビティを生成
    Activity act1, act2, act3;
    act1 = new Activity("atp://kazu.yamato.ibm.com:434/", "JMT.s
samples.TextFieldAgent", new Message("request"));
5    act2 = new Activity("atp://" + _home.getHost() + ":600", "JMT.s
amples.TextFieldAgent", new Message("request"));
    act3 = new Activity("atp://" + _home.getHost() + ":434", "JMT.s
amples.ReportAgent", new Message("report"));

10    //プランにアクティビティとテンプレートを追加
    try{
        Vector predecessorIndexList;
        predecessorIndexList = new Vector();
        //プランにANDSplitを追加
15        Integer hwd1 = plan.addNode(null, new ANDSplit());
        predecessorIndexList.addElement(hwd1);
        //ANDSplitのサクセッサとして3つのアクティビティを追加
        Integer hwd2 = plan.addNode(predecessorIndexList, act1);
        Integer hwd3 = plan.addNode(predecessorIndexList, act2);
20        //ANDJoinのプレデセッサ・リストに3つのアクティビティのインデ
        ックスを追加
        predecessorIndexList.removeAllElements();
        predecessorIndexList.addElement(hwd2);
        predecessorIndexList.addElement(hwd3);
25        //プランにORJoinを追加
        Integer hwd4 = plan.addNode(predecessorIndexList, new OR
```

```
Join());  
    predecessorIndexList.removeAllElements();  
    predecessorIndexList.addElement(hwd4);  
    //ORJoinのサクセッサとしてアクティビティを追加  
5    Integer hwd5 = plan.addNode(predecessorIndexList, act3);  
    } catch (BlockedByIteratorException e) {  
        Utility.print(this, e.getMessage());  
    }  
    //モデレータ・エージェントを生成  
10    createModeratorAgent(plan);  
}  
  
private void createModeratorAgent(Plan plan) {  
    try {  
15    getAgletContext().createAglet(null, "JMT.ModeratorAgent", (Object)plan);  
    } catch (AgletException e) {  
        Utility.print(this, e.getMessage());  
    } catch (IOException e) {  
20    Utility.print(this, e.getMessage());  
    }  
    }  
  
    private PlanID generatePlanID() {  
    //プランIDを生成  
25    PlanID planID = new PlanID(_home.toString(), _id.toString()  
        +(new Integer(_planIndex)).toString
```

```
());  
    _planIndex++;  
    return planID;  
}  
5    }
```

F. プランの実行

F-1. Chainテンプレートを含むトポロジー (図14) の場合

10

Chainテンプレートを含むトポロジーの場合における移動オブジェクト群の動作を図17乃至図21を使用して説明する。図17は、プランの実行手順を示すフローチャートであり、図18は、モデレータ・テンプレートの機能ブロック図である。図21は、移動オブジェクト群がオリジンのプレースからディスティネーションのプレースに移動する状態の概念図である。まず、図17を参照する。モデレータ・エージェント351が生成され、プラン353が割り当てられると(ブロック843)、モデレータ・エージェントはプランを自動的に実行する(ブロック845)。

15

20

モデレータ・エージェント351からプラン・オブジェクト353に対し送出された、"NextStep"のインストラクションに応答して、プラン・オブジェクト353はまず、カレントのプランノードを取り出す(ブロック847)。本発明の好適な実施例においては、プラン・イタレータオブジェクト352が現在プランノードのどのステップにあるかを管理しており、プラン・オブジェクト353の問い合わせに応答して、現在

25

プランノードのどのステップにいるか(カレント)の情報を提供する。

カレントの問い合わせに対し、プラン・イタレータオブジェクト 3 5 2 が " 0 " (最初のステップ) を返した場合、プラン・オブジェクト 3 5 3 は、a c t 1 を取り出す。そして、プラン・オブジェクト 3 5 3 は、a c t 1 に対して、その中で定義されているアクティビティにアクティビティが管理しているリクエストを実行することを依頼する (ブロック 8 4 9) 。

アクティビティは内部に保持しているエージェントのクラス名 (エージェント名) が同一のプレースに存在しているか否か判断する。図 1 9 に示すように、モデレータ・エージェント 3 5 0 はプレース 3 6 0 に到達すると、プレース 3 6 0 に対し自分のエージェント名を教え、プレース 3 6 0 はこれをエージェントリスト 3 6 9 として管理している。そして、プレース 3 6 0 に存在するエージェントは、プレース 3 6 0 に対し "getAgentList" のインストラクションを出し、問い合わせを行うと、プレースは AgentList を返す機能があるので、同一プレースに存在するエージェントを知ることができる。

アクティビティは見つけたエージェント (アクター・エージェント) 3 6 1 に対し内部に保持しているメッセージ (リクエスト) を送る。アクター・エージェント 3 6 1 はこのリクエストに対し、リターンオブジェクトをリクエスト側のエージェント 3 5 0 に送出する。リクエスト側のエージェントは、この受け取ったオブジェクトを R e s u l t のレコード 5 2 5 (図 1 4) に格納する。

アクティビティはプラン・オブジェクト 3 5 3 に依頼の仕事が終わったことを知らせ、これに応答してプラン・オブジェクト 3 5 3 は a c t 1 に処理が終了したことを示すマークを付ける (ブロック 8 5 5) 。そして、プラン・オブジェクト 3 5 3 はプラン・イタレータ 3 5 2 に対し、ステップを進めることを依頼し (ブロック 8 5 9) 、プラン・イタレー

タ 3 5 2 は、プラン・オブジェクト 3 5 3 にカレントの情報を提供する。

この場合、カレントは"1"(2番目のステップ)であるので、プラン・オブジェクトはChainテンプレートのオブジェクトを取り出す（ブロック 8 4 5、8 4 7）。そして、プラン・オブジェクトはChain
5 に対して、実行を依頼する（ブロック 8 4 9）。

Chainのモデレーション管理マネジャ 6 0 3 は、ステート管理部 6 0 1 に現在のステートを問い合わせる。ステート管理部 6 0 1 は、デフォルトでオリジンの状態になっているので、オリジンであることをモデレーション管理マネジャ 6 0 3 に知らせる。モデレーション管理マネ
10 ジャ 6 0 3 は、ステート管理部 6 0 1 から、ステートの情報を得ると、ステートの切り換えをステート管理部 6 0 1 に依頼する。ステート管理部 6 0 1 は、これに応答してステートをオリジンからディスティネーションに切り換える。

このテンプレートの行うステートの切り換え及びオリジンまたはディスティネーションにおける処理のフローを図 2 0 に示す。このフローチャートは複数種類存在するテンプレートに共通な動作を示しており、ブロック 8 2 5 7 のオリジンにおける処理とブロック 8 3 1 のディスティネーションにおける処理の内容がテンプレートの種類によって異なる処理が実施される。
15

モデレーション管理マネジャ 6 0 3 は、オリジン用モデレーション・モジュール 6 0 5 に処理を依頼する。オリジン用モデレーション・モジュール 6 0 5 とディスティネーション用モデレーション・モジュール 6 0 7 には、夫々のプレースにおいてテンプレートが動作するためのインストラクションのセットが管理されている。
20

オリジン用モデレーション・モジュール 6 0 5 は、モデレータ・エー

ジェント参照モジュール 5 9 9 に問い合わせを行うことにより、現在モデレータ・エージェント 3 5 1 の存在するプレースのアドレスを取得する。

5 また、サクセッサ・リスト 6 0 9 に対し、サクセッサ・リスト 6 0 9 に登録されているアクティビティのアドレスの渡すように依頼する。サクセッサ・リスト 6 0 9 は、登録されているアクティビティ (a c t 2) からアドレスを取得し、オリジン用モデレーション・モジュール 6 0 3 に渡す。

10 オリジン用モデレーション・モジュール 6 0 3 は、アクティビティのアドレスと現在モデレータ・エージェントの存在するプレースのアドレスを比較し、もし 2 つが異なっていたらアクティビティのアドレスにモデレータ・エージェントを移動させる。具体的には、オリジン用モデレーション・モジュール 6 0 5 は、モデレータ・エージェント参照モジュールを介して、移動を命令するインストラクションを送出する。同一アドレスの場合には、処理を終了する。

15 プラン・オブジェクトは移動先で再びプラン・オブジェクトに対し 1 ステップの実行を依頼する。同様にカレントのプランノードを取り出す。プラン・オブジェクトは C h a i n に対し、次のステップの実行を依頼する。C h a i n はステートの情報を確認し、ディスティネーションの処理を行う。ステートの情報をオリジンの状態に戻す。

20 プランは C h a i n のディスティネーションの処理の終了を確認すると、次のカレントのプランノードを取り出し (a c t 2)、アクティビティの実行を依頼する。a c t 2 は、a c t 1 が内部に保持する r e s u l t を取得する。

25 本発明の好適な実施例においては、ディスティネーションモジュール 6 0 7 がプレデセッサ・リスト 5 9 1 を参照し、C h a i n に対し、処

理結果の送出を依頼する"getResult"のインストラクションを送出する。
Chainは、これに応答し、自己の保有するResultを探索する。
しかし、ChainはResultを保有していないので、さらにプレ
デセッサ・リスト591を参照し、プレデセッサ・リスト591に登録
5 されたオブジェクト(act1)にResultをもらいに行き、その
Resultをact2に返す。プレデセッサ・リスト591に登録さ
れたオブジェクトが存在しない場合には、act2にResultを保
有していないことを示すDummyを送出する。この処理はテンプレ
10 トにResultを格納するレコードを作成することによって代替可能
である。

act2は、内部に保持しているエージェントのクラス名をキーに同
一スペースに存在するレポートエージェントの参照を得る。act2は
レポートエージェントに対し内部に保持しているメッセージをact1
から入手したresultを引き数にして送る。プランは次のカレント・
15 ノードを取り出す。この場合、カレント・ノードは存在しないので、プ
ランはこれを検出して処理を終了する。ディスティネーション用モデ
レーション・モジュール607はモデレータ・エージェント参照モジュ
ール599を介して、モデレータ・エージェントにモデレータ・エージェ
ントを消滅させるインストラクションを送出する。

20 F-2. Iterationテンプレートを含むトポロジー (図15)
の場合

図22は、Iterationテンプレートによって制御される移動
オブジェクト群の動作の概要を示す図である。以下、図18を参照し、
25 Iterationテンプレートを含むトポロジーの場合の処理を説明
する。Chainの場合と同様に、モデレータ・エージェント351を

生成し、その際の引き数として作成したプラン・オブジェクト 353 を渡し、プラン 353 を 1 ステップずつ実行する。また、プラン・オブジェクト 353 は、Chain の場合と同様に、カレントのプランノード (Chain) を取り出し、Chain はオリジンにおける処理を実行する。また、移動先において、「F-1. Chain テンプレートを含むトポロジーの場合」で説明したように、ディスティネーションにおける処理の実行、act 1 のアクティビティの実行を行う。

act 1 の処理が終了すると、次にプラン 353 は Iteration に次のステップの実行を依頼する。図 23 は、Iteration テンプレートのオリジンにおける処理手順を示すフローチャートである。Iteration テンプレートのオリジンにおける処理においては、まず、次のステップに進んでよいか、例えば、act 1 の result の情報が所定の条件に一致しているか否かチェックを行う (ブロック 873)。

もし条件を満たしている場合、内部に保持するサクセッサ情報をサクセッサ・リストに格納する (ブロック 875)。そして、サクセッサ・リストの act 2 よりアドレスを取り出し (ブロック 877)、そのアドレスにモデレータ・エージェントを移動させる (ブロック 879)。もし、所定の条件に一致していない場合には、ステートをオリジンにし (ブロック 881)、リターン・ノード・インデックスよりプランノードの参照を得、そのプランノードでサクセッサ・リストの要素を置き換え、次のステップに進む (ブロック 883)。以下「F-1. Chain テンプレートを含むトポロジーの場合」において説明したように act 2 の処理を行い、処理を終了する。

F-3. AND Split 及び AND Join テンプレートを含むトポ

ロジー（図16）の場合

図24、25は、ANDSplit、ANDJoinテンプレートによって制御される移動オブジェクト群の動作の概要を示す図である。以下、図18を参照し、ANDSplit及びANDJoinテンプレートを含むトポロジーの場合の処理を説明する。Chainの場合と同様に、モデレータ・エージェント351を生成し、その際の引き数として作成したプラン・オブジェクト353を渡し、プラン353は、1ステップずつ実行される。

また、プラン・オブジェクト353は、Chainの場合と同様に、カレントのプランノードであるANDSplitを取り出す。プラン353はANDSplitに対して次のステップの実行を依頼する。図26は、ANDSplitテンプレートにおけるオリジンの処理の処理手順を示すフローチャートである。

モデレーション管理マネージャ603は、ステート管理部601のステート情報を検索し、デフォルトである、「オリジン」の情報を取得する。モデレーション管理マネージャ603は、オリジン用モデレーション・モジュール605にオリジンにおける処理を行うよう依頼する。モデレーション管理マネージャ603は、ステートをディスティネーションに変えるようステート管理部に依頼する（ブロック893）。そして、ANDSplitは、サクセッサ・リストを参照し（ブロック897）、サクセッサ・リストの最初のアクティビティを取り出し、マークドインデックス595に"0"をセットする（ブロック899）。

次に、モデレーション管理マネージャ603は、モデレータ・エージェントのクローンを作成するようモデレータ・エージェント参照モジュール599を介してモデレータ・エージェント351に依頼する（ブロッ

ク 9 0 1。モデレータ・エージェントは、この依頼に回答して、モデレータ・エージェント及びモデレータ・エージェントの保持するプラン、テンプレート、アクティビティ等、全てのオブジェクトのコピーも作成する。

5 コピーによって作成されたクローンのモデレータ・エージェントは、クローンの作成の完了に回答して、自分のエージェントIDをモデレータ・エージェント参照モジュール 5 9 9 に教える。モデレータ・エージェント参照モジュール 5 9 9 は、クローンの作成が完了したことをオリジン用モジュレーションモジュール 6 0 5 に知らせる。

10 オリジン用モジュレーションモジュール 6 0 5 はこれに回答して、マークドインデックスの値 (0) に対応するアクティビティ (a c t 1) のアドレスをサクセッサ・リスト 6 0 9 から取得する。オリジン用モデレータモジュール 6 0 5 は、モデレータ・エージェント参照モジュール 5 9 9 を介して、クローンを a c t 1 のアドレスに対応するスペースに移動することをモデレータ・エージェント 3 5 1 に依頼する (ブロック 9 0 3) 。

15 オリジン用モジュレーションモジュール 6 0 5 は、サクセッサ・リスト 6 0 9 を参照し、次のアクティビティが登録されているか否か判断し、登録されている場合には、さらに、もう一つのクローンの作成の処理を同様の手順で行い、a c t 2 のスペースに移動させる (ブロック 8 9 7 ~ 9 0 3) 。この時、a c t 2 に移動させるクローンのマークドインデックス 5 9 5 は a c t 1 に移動するクローンと異なり、"1" にしておく (ブロック 8 9 9) 。

20 a c t 2 に移動するクローンの作成及び移動の処理が終了すると、オリジン用モジュレーションモジュール 6 0 5 は、サクセッサ・リスト 6 0 9 を参照し、次のアクティビティが登録されているか否か判断する (

ブロック 897)。この例の場合には、サクセッサ・リスト 609 には、もう登録されているアクティビティは存在しないので、オリジン用モデレーション・モジュール 605 は、クローンの作成及び移動の処理が終了したことを認識する。

5 オリジン用モデレーション・モジュール 605 はこれに応答して、モデレータ・エージェント参照モジュール 599 を介して、もとの場所（オリジンのプレース）に残っているモデレータ・エージェント 351 を消去するインストラクションをモデレータ・エージェント 351 に送出する（ブロック 905）。

10 この一方、クローンのモデレータ・エージェントは、夫々のプレースにおいて、次のプランステップを実行する。このとき、カレント・ノードとして AND Split が再び取り出される。この次のカレントを決定するロジックのフローチャートを図 27 に示す。モデレーション管理マネージャ 603 は、ステート管理部 601 を参照し、ステート情報を取得し、現在のステートはディスティネーションであることを認識し、ディスティネーション用モデレーション 607 に処理の実行を依頼する。

15 モデレーション管理マネージャ 603 はステートの情報をオリジンの状態に戻す。プランは AND Split のディスティネーションの処理の終了を確認すると、マークドインデックスに対応したプランノードを取り出し（act 1 または act 2）、アクティビティの実行を依頼する。

20

 プラン・オブジェクト 353 はプラン・イタレータ 352 を参照して、次のカレントのノードを取り出す。このとき、プラン・イタレータは、サクセッサ・リスト 609 及びマークドインデックス 595 を参照し、登録されている値を検査する（図 27）。

25

 act 1、act 2 は、先に実行されたアクティビティが保持する R

resultを取得する。この例では、ANDSplitテンプレートの前にはアクティビティが実行されていないため、act1、act2は、先に実行されたアクティビティが保持するResultを、結果的に取得しないことになる。

- 5 まず、act1に対応するクローンのエージェントにおいて、次のカレント・ノードとしてANDJoinが選ばれる。プラン・オブジェクト353は、ANDJoinテンプレートのモデレータ管理マネージャ603に次のステップの実行を依頼する。モデレータ管理マネージャ603は、ステート管理部601の情報から、現在オリジンの処理を行うべきことを認識し、オリジン用モデレーション・モジュール605に実行要求を出す。また、ステート管理部601にステートの変更を依頼する。

- 10 オリジン用モデレーション・モジュール605はサクセッサ・リスト609を参照し、サクセッサ・リスト609に登録されたアクティビティ(act3)のアドレス情報を取得する。そして、モデレータ・エージェント参照モジュール599を介してモデレータ・エージェント351に入手したアドレス情報に対応したブレースへの移動を指示する。

- 15 act2に対応するクローンのエージェントもact1の場合と同様に処理を行い、act3のブレースに移動する。2つのモデレータ・エージェントは新たなブレースに到着したので、それぞれ次のステップを実行する。プラン・オブジェクト353は、カレント・ノードとしてANDJoinテンプレートをとり出し、次のステップを実行する。

- 20 図28、29は、ANDJoinテンプレートのディスティネーションにおける処理手順を示すフローチャートである。ANDJoinのモデレーション管理マネージャ603は、ステート管理部601を参照して、現在のステートがディスティネーションであることを判断し、ディスティネーション用モデレーション・モジュール607に実行を依頼する。

各モデレーション管理マネジャ603は、ステートをオリジンの状態に戻す。a c t 1に対応するクローンのディスティネーション用モデレーション・モジュール607は、プレデセッサ・リスト591を参照し、終了のマークのついたアクティビティ（この場合a c t 1）を取り出す（ブロック913）。この取り出されたアクティビティがプレデセッサ・インデックス・リストの中で一番小さなインデックスを有しているか否か検査し（ブロック915）、一番小さなインデックスを有していると判断された場合には、他のクローンエージェントのR e s u l tを受け取る処理を行う。

具体的には、ディスティネーション用モデレーション・モジュール607は、同一スペース上エージェント参照モジュール597に同一スペースに存在するエージェントリストをスペースから取得することを依頼する。同一スペース上エージェント参照モジュール597はこれに応答して、スペース360に対し"getAgentList"のインストラクションを出し、問い合わせを行う。スペースはAgentListを返すことによって、同一スペースに存在するエージェントのリストを入手する。

同一スペース上エージェント参照モジュール597はこのリストをディスティネーション用モデレーション・モジュール607に渡す。ディスティネーション用モデレーション・モジュール607は、このリストから同一のプランIDを有するエージェントを探す（ブロック921）。

同一のプランIDを有するエージェントが発見された場合、そのエージェントのプランを取得する。そして、取得したプランのカレント・ノードを調べ（ブロック929）、同じANDJoinであれば、そのエージェントのR e s u l tを受け取る処理を行う。本発明の好適な実施例においては、さらに、カレント・ノード・インデックスが同じか否か

判断し（ブロック 9 3 1、9 3 3）、同じ場合のみ融合の処理に入る。
これは、AND Join の処理対象を誤認することを防止するためである。
このとき、ディスティネーション用モデレーション・モジュール 6
0 7 は、プレデセッサ・リストを参照し、プレデセッサが幾つあるかを
5 把握し、その数に対応して R e s u l t を受け取る処理を行う。

具体的には、ディスティネーション用モデレーション・モジュール 6
0 7 は、受領したプランのプレデセッサ・リスト 5 9 1 を参照し、終了
のマークのついたアクティビティを取り出し、そのアクティビティの保
有している R e s u l t を受け取る。そして、この R e s u l t を対応
10 するアクティビティの R e s u l t にセットする。プランノード自体を
置き換える処理を行ってもよい（ブロック 9 3 5）。図 3 0 は、この R
e s u l t を受け取る処理の概要を示す図である。

ディスティネーション用モデレーションエージェント 6 0 7 は、R e
s u l t のセットが終了したことを検出した後、モデレータ・エージェ
15 ント参照モジュールを介して、相手のモデレータ・エージェント 3 5 1
に対して消去を指示するインストラクションを送出する（ブロック 9 3
7）。相手のモデレータ・エージェントは、このインストラクションを
受け、モデレータ・エージェント及び、自分の管理するオブジェクト（
プラン、テンプレート、アクティビティ等）を消去する。

この一方、a c t 2 に対応するエージェントは、新たなプレースに到
着したことに伴い、次のステップを実行する。このエージェントのディ
スティネーション用モデレーション・モジュール 6 0 7 も、プレデセッ
サ・リスト 5 9 1 を参照し、終了のマークのついたアクティビティ（こ
の場合 a c t 2）を取り出す（ブロック 9 1 3）。この取り出されたア
クティビティがプレデセッサ・インデックス・リストの中で一番小さな
25 インデックスを有しているか否か検査し（ブロック 9 1 5）、一番小

なインデックスでないと判断される。この場合には、他のクローンエージェントの `Result` を渡すのを待つため `Sleep` 状態に入る（ブロック 925）。

再び、話を `act1` に対応するクローンに戻すと、ディスティネーション用モデレーション・モジュール 607 は、プレデセッサ・リストの数に対応して（プレデセッサ・リストの要素数 - 1 回）`Result` を受け取り、クローンの消去を行う処理を完了した後、プランは、次のカレントのプランノードを取り出し（`act3`）、アクティビティの実行を依頼する。

`act3` は、`act1`、`act2` が内部に保持する `result` を取得する。本発明の好適な実施例においては、ディスティネーションモジュール 607 がプレデセッサ・リスト 591 を参照し、（`act1` に対応するクローンの）`act1`、`act2` に対し、処理結果の送出を依頼する "getResult" のインストラクションを送出する。`act1`、`act2` は、これに応答し、自己の保有する `Result` を送出する。

`act3` は、内部に保持しているエージェントのクラス名をキーに同一スペースに存在するエージェント（この例ではレポートエージェント）の参照を得る。`act3` はレポートエージェントに対し内部に保持しているメッセージを `act1`、`act2` から入手した `result` を引き数にして送る。

プランは次のカレント・ノードを取り出す。この場合、カレント・ノードは存在しないので、プランはこれを検出して処理を終了する。そして、モデレータ・エージェントは、処理の終了を検出しモデレータエージェント及び自己が管理するオブジェクトの消去をして処理を終了する。

F-4. `ORSplit` テンプレートを含むトポロジーの場合

図31は、OR Splitテンプレートによって制御される移動オブジェクト群の動作の概要を示す図である。OR Splitの場合、ChainやAND Splitの場合と異なりオリジンの処理において図32に示す処理を行う。OR Splitテンプレートは一つのプレデセッサのアクティビティと複数のサクセッサのアクティビティを有している。オリジナルのプレースにおいては、"chooseActivity"というメソッドが呼び出されることによってサクセッサのリストから一つのアクティビティが選択され、その選択されたアクティビティのみが実行される。

OR Splitのオリジンの処理においては、まず、ステートがディスティネーションに変更される（ブロック982）。そして、"chooseActivity"関数をコールし、所定の条件に合致する実行すべきアクティビティを選択する（ブロック983）。そして、そのアクティビティのインデックスをマークインデックスにする（ブロック984）。そして、選択されたアクティビティのアドレスを入手し、モデレータ・エージェント351に該アドレスへの移動を要求する（ブロック985）。

F-5. OR Joinテンプレートを含むトポロジーの場合

図33は、OR Joinテンプレートによって制御される移動オブジェクト群の動作の概要を示す図である。OR Joinの場合、ChainやAND Joinの場合と異なりディスティネーションの処理において図34に示すような処理を行う。OR Joinテンプレートは、複数のプレデセッサのアクティビティと一つのサクセッサのアクティビティを有している。OR Joinは、ディスティネーションにおいて、最も早く到着したものがDisposeエージェントを生成し、他の移動エ

ージェントを消去する処理を行う。

まず、最も早くディスティネーション 690 に到着した移動オブジェクト群の ORJoin テンプレート 694 は、"findDisposeAgent" のメソッドを送出して、同一スペース上に Dispose エージェントが存在しているか否か判断する（ブロック 943）。存在しない場合は、他のエージェントを消去する Dispose エージェントを生成する（ブロック 945）。このとき引き数としてプラン及びプレデッサ・リストのアクティビティの数 - 1 を渡す。Dispose エージェントは、プレデッサ・リストのアクティビティの数 - 1 の移動オブジェクト群を待ち受け消去し、役目が終了すると自分自身を消去する。本発明の好適な実施例においては、最も早くディスティネーション 690 に到着したか否かを生き残りの条件としているが、Result の内容を条件として、継続処理するものを決定してもよい。

この一方、他の移動オブジェクト群の ORJoin テンプレート 696 は、同様に "findDisposeAgent" のメソッドを送出して、同一スペース上に Dispose エージェントが存在しているか否か判断する（ブロック 943）。この場合、既に Dispose エージェントが存在しているので、Dispose エージェントに消去される処理に入る。具体的には、Dispose エージェントが同じプランを保持するか（ブロック 947）、また、同じカレント・ノード・インデックスか（ブロック 949）判断し、自分が消去されるべき、Dispose エージェントであると確認する。

自分が消去されるべき、Dispose エージェントであると確認できた場合には、Dispose エージェントに "countdown" のメッセージを送り、Dispose エージェントの処理数を減らす（ブロック 951）。そして、ORJoin テンプレート 696 は、モデレータ・エ

エージェント692にdispose要求を送出し、モデレータ・エージェント692を消滅させる。

その他、本発明の好適な実施例においては、複数のプレデセッサのアクティビティと、プレデセッサのアクティビティよりも少ない数の複数のサクセッサのアクティビティを有する一部Joinテンプレート等も提供されているが、これらのテンプレートは上述のテンプレートの組合せでも実施可能であるため説明を省略する。例えば、一部Joinテンプレートは複数のプレデセッサのアクティビティと、1つのサクセッサのアクティビティとをリンクするORJoinテンプレートと、所望の数のORJoinテンプレートが実行されたことを判断するアクティビティと、このアクティビティが所望の条件をクリアしたか否かの判断によってORJoinテンプレートに戻るIterationテンプレートと、ダミーのアクティビティと、所望の数に分裂させるANDSplitの組合せによって実施可能である。

[産業上の利用可能性]

以上説明したように、本発明によれば、移動エージェントの実行時において、他の資源の占有を最小限に抑えることができる。

また、本発明によれば、効率的な処理を行うことにより、ユーザが移動エージェントを放ってから結果が返ってくるまでに要する処理時間を短縮することができる。

さらに、本発明によれば、高度なプログラミングの知識を必要とせず、プログラマの開発労力及び開発時間の負担が少ない移動エージェントの開発が可能となる。

そして、本発明によれば、開発者が直感的にその機能を把握でき、親

しみやすい移動エージェント開発用プログラム部品を提供することができる。

請求の範囲

- 5 1. 移動モデレータ・エージェントの制御の下、第1のプレースから第2のプレースへ移動し、前記第1のプレースにおいて第1のアクティビティを実行し、前記第2のプレースにおいて第2のアクティビティを実行する移動オブジェクト群の一部を構成するオブジェクトであって、
- (a) 前記第1のアクティビティとのリンク情報を管理するプレデセッサ・リストと、
- 10 (b) 前記第2のアクティビティとのリンク情報を管理するサクセッサ・リストと、
- (c) 前記第1のプレースにおいて、前記移動モデレータ・エージェントに前記第2のプレースへの移動を要求するモデレーション・モジュールと、
- 15 を含むオブジェクト。
2. 移動モデレータ・エージェントの制御の下、第1のプレースから第2のプレースへ移動し、前記第1のプレースにおいて第1のアクティビティを実行し、前記第2のプレースにおいて第2のアクティビティを実行する移動オブジェクト群の一部を構成するオブジェクトであって、
- 20 (a) 前記第1のアクティビティとのリンク情報を管理するプレデセッサ・リストと、
- (b) 前記第2のアクティビティとのリンク情報を管理するサクセッサ・リストと、
- (c) 前記モデレータモジュールが前記第1のプレースに存在しているのか、前記第2のプレースに存在しているのを判断するステート管理部と、
- 25

(d) 前記第1のプレースにおいて、前記移動モデレータ・エージェントに前記第2のプレースへの移動を要求するモデレーション・モジュールと、

を含む移動オブジェクト。

- 5 3. 移動モデレータ・エージェントの制御の下、第1のプレースから第2のプレースへ移動し、前記第1のプレースにおいて第1のアクティビティを実行し、前記第2のプレースにおいて第2のアクティビティを実行する移動オブジェクト群であって、

10 (a-1) 前記第1のアクティビティとのリンク情報を管理するプレデセッサ・リストと、

 (a-2) 前記第2のアクティビティとのリンク情報を管理するサクセッサ・リストと、

15 (a-3) 前記モデレータモジュールが前記第1のプレースに存在しているのか、前記第2のプレースに存在しているのかを判断するステート管理部と、

 (a-4) 前記第1のプレースにおいて、前記移動モデレータ・エージェントに前記第2のプレースへ前記移動オブジェクト群の移動を要求するモデレーション・モジュールと、

 を含むモデレータ・テンプレートと、

20 (b) 前記第1のアクティビティ、前記第2のアクティビティ、前記モデレータ・テンプレートのいずれが現在実行権限を有しているかを判断するプラン・イタレータと、

25 (c) 前記第1のアクティビティ、前記第2のアクティビティ、前記モデレータ・テンプレートの実行順序を管理するプラン・オブジェクトと、

 を含む移動オブジェクト群。

4. 移動モデレータ・エージェントの制御の下、第1のアドレスによって特定される第1のプレースと第2のアドレスによって特定される第2のプレースへ移動し、前記第1のプレースにおいて第1のアクティビティを実行し、前記第2のプレースにおいて第2のアクティビティを実行する移動オブジェクト群を制御する方法であって、

(a) 前記移動モデレータ・エージェントに前記第1のアクティビティへのリンク情報を含む第1の移動モデレータ・エージェントと前記第2のアクティビティへのリンク情報を含む第2の移動モデレータ・エージェントの用意を要求する段階と、

(b) 前記第1のアドレスを入手する段階と、

(c) 前記第2のアドレスを入手する段階と、

(d) 前記第1の移動モデレータ・エージェントに前記第1のプレースに移動することを要求する段階と、

(e) 前記第2の移動モデレータ・エージェントに前記第2のプレースに移動することを要求する段階と、

を含む移動オブジェクト群の制御方法。

5. 移動モデレータ・エージェントの制御の下、第1のアドレスによって特定される第1のプレースと第2のアドレスによって特定される第2のプレースと第3のアドレスによって特定される第3のプレースへ移動し、前記第1のプレースにおいて第1のアクティビティを実行し、前記第2のプレースにおいて第2のアクティビティを実行し、前記第3のプレースにおいて第3のアクティビティを実行する移動オブジェクト群を制御する方法であって、

(a) 前記第1のプレースにおいて、前記移動モデレータ・エージェントに前記第2のアクティビティへのリンク情報を含む第1の移動モデレータ・エージェントと前記第3のアクティビティへのリンク情報を含

む第2の移動モデレータ・エージェントの用意を要求する段階と、

(b) 前記第2のアドレスを入手する段階と、

(c) 前記第3のアドレスを入手する段階と、

5 (d) 前記第1の移動モデレータ・エージェントに、前記第1のプレースから前記第2のプレースに移動することを要求する段階と、

(e) 前記第2の移動モデレータ・エージェントに、前記第1のプレースから前記第3のプレースに移動することを要求する段階と、

(f) 前記第2のプレースにおいて前記第2のアクティビティを実行することを要求する段階と、

10 (g) 前記第3のプレースにおいて前記第3のアクティビティを実行することを要求する段階と、

を含む移動オブジェクト群の制御方法。

6. 移動モデレータ・エージェントの制御の下、複数のプレースを移動し、プレースに対応したアクティビティを実行する移動オブジェクト群を制御する方法であって、

15 (a) 第1のプレースにおいて、第1の移動モデレータ・エージェントにリンクした第1のアクティビティを実行し、第1の処理結果を得る段階と、

20 (b) 前記第1のプレースに存在し、前記第1の処理結果を保持する前記第1の移動モデレータ・エージェントに対し第2のプレースに移動することを要求する段階と、

(c) 第3のプレースにおいて、第2の移動モデレータ・エージェントにリンクした第2のアクティビティを実行し、第2の処理結果を得る段階と、

25 (d) 前記第3のプレースに存在し、前記第2の処理結果を保持する前記第2の移動モデレータ・エージェントに対し前記第2のプレースに

移動することを要求する段階と、

(e) 前記第2のプレースにおいて、前記第2の移動モデレータ・エージェントに対し前記第2の処理結果を前記第1の移動モデレータ・エージェントに渡すことを要求する段階と、

5 (f) 前記第2のプレースにおいて、前記第1の移動モデレータ・エージェントが前記第1の処理結果とともに前記第2の処理結果を保持する段階と、

を含む移動オブジェクト群の制御方法。

7. 移動モデレータ・エージェントの制御の下、複数のプレースを移動
10 し、プレースに対応したアクティビティを実行する移動オブジェクトを制御する方法であって、

(a) 第1のプレースにおいて、エージェント識別情報を有する第1の移動モデレータ・エージェントにリンクした第1のアクティビティを実行し、第1の処理結果を得る段階と、

15 (b) 前記第1のプレースに存在し、前記第1の処理結果を保持する前記第1の移動モデレータ・エージェントに対し第2のプレースに移動することを要求する段階と、

20 (c) 第3のプレースにおいて、前記第1の移動モデレータ・エージェントの有するエージェント識別情報と同一の識別情報を有する第2の移動モデレータ・エージェントにリンクした第2のアクティビティを実行し、第2の処理結果を得る段階と、

(d) 前記第3のプレースに存在し、前記第2の処理結果を保持する前記第2の移動モデレータ・エージェントに対し前記第2のプレースに移動することを要求する段階と、

25 (e) 前記第2のプレースにおいて、前記第2の移動モデレータ・エージェントに対し前記第2の処理結果を前記第1の移動モデレータ・エ

ージェントに渡すことを要求する段階と、

(f) 前記第2のプレースにおいて、前記エージェント識別情報と同一の識別情報を有する移動モデレータ・エージェントを探索する段階と、

5 (g) 前記第2のプレースにおいて、前記第1の移動モデレータ・エージェントが前記第1の処理結果とともに前記第2の処理結果を保持する段階と、

を含む移動オブジェクトの制御方法。

10 8. 第1のプレースから第2のプレースへ移動し、前記第1のプレースにおいて第1のアクティビティを実行し、前記第2のプレースにおいて第2のアクティビティを実行する移動オブジェクト群であって、

(a) 前記第1のアクティビティ、前記第2のアクティビティの実行順序を管理するプラン・オブジェクトと、

15 (b) 前記第1のアクティビティ、前記第2のアクティビティのいずれが現在実行権限を有しているかを判断するプラン・イタレータと、

を含む移動オブジェクト群。

20 9. 移動モデレータ・エージェントの制御の下、第1のプレースから第2のプレースへ移動し、前記第1のプレースにおいて第1のアクティビティを実行し、前記第2のプレースにおいて第2のアクティビティを実行する移動オブジェクト群であって、前記第1のプレースにおいて、前記第1のアクティビティと前記第2のアクティビティが前記移動モデレータ・エージェントに前記第2のプレースへ前記移動オブジェクト群の移動を要求するモデレータ・テンプレートにリンクされている移動オブジェクト群を生成方法であって、

25 (a) 前記第1のアクティビティにおいて、前記モデレータ・テンプレートとのリンク情報を管理するサクセッサ・リストを生成する段階

と、

5 (b) 前記モデレータ・テンプレートにおいて、前記第1のアクティビティとのリンク情報を管理するプレデセッサ・リストと、前記第2のアクティビティとのリンク情報を管理するサクセッサ・リストとを生成する段階と、

 (c) 前記第2のアクティビティにおいて、前記モデレータ・テンプレートとのリンク情報を管理するプレデセッサ・リストを生成する段階と、

 を含む方法。

10 10. 第1のプレースにおいて実行される第1のアクティビティと、第2のプレースにおいて実行される第2のアクティビティと、前記第1のプレースにおいて移動オブジェクト群の移動を制御する移動モデレータ・エージェントに前記第2のプレースへ移動することを要求するモデレータ・テンプレートを含む移動オブジェクト群を生成方法であって、

15 (a) 前記第1のアクティビティと、前記モデレータ・テンプレートと、前記第2のアクティビティと、の実行順序を管理する順序リストを生成する段階と、

 (b) 前記第1のアクティビティにおいて、前記モデレータ・テンプレートとのリンク情報を管理するサクセッサ・リストを生成する段階と、

20 (c) 前記モデレータ・テンプレートにおいて、前記第1のアクティビティとのリンク情報を管理するプレデセッサ・リストと、前記第2のアクティビティとのリンク情報を管理するサクセッサ・リストとを生成する段階と、

25 (d) 前記第2のアクティビティにおいて、前記モデレータ・テンプレートとのリンク情報を管理するプレデセッサ・リストを生成する段階

階と、

を含む方法。

1 1. 第1のプレイスにおいて実行される第1のアクティビティと、第
2のプレイスにおいて実行される第2のアクティビティと、前記第1の
5 プレイスにおいて移動オブジェクト群の移動を制御する移動モデレータ・
エージェントに前記第2のプレイスへ移動することを要求するモデレー
タ・テンプレートを含む移動オブジェクト群を生成する装置であって、

(a) 前記第1のアクティビティにおいて、前記モデレータ・テン
プレートとのリンク情報を管理するサクセッサ・リストを生成する手段
10 と、

(b) 前記モデレータ・テンプレートにおいて、前記第1のアクテ
ィビティとのリンク情報を管理するプレデセッサ・リストと、前記第2
のアクティビティとのリンク情報を管理するサクセッサ・リストとを生
成する手段と、

15 (c) 前記第2のアクティビティにおいて、前記モデレータ・テン
プレートとのリンク情報を管理するプレデセッサ・リストを生成する手
段と、

を含む装置。

20 1 2. 第1のプレイスにおいて実行される第1のアクティビティと、第
2のプレイスにおいて実行される第2のアクティビティと、前記第1の
プレイスにおいて移動オブジェクト群の移動を制御する移動モデレータ・
エージェントに前記第2のプレイスへ移動することを要求するモデレー
タ・テンプレートを含む移動オブジェクト群を生成するプログラムを格
納するコンピュータによって読み取り可能な記憶媒体であって、

25 該プログラムは、

(a) 前記第1のアクティビティにおいて、前記モデレータ・テン

プレートとのリンク情報を管理するサクセッサ・リストを生成することを前記コンピュータに指示するプログラムコード手段と、

5 (b) 前記モデレータ・テンプレートにおいて、前記第1のアクティビティとのリンク情報を管理するプレデセッサ・リストと、前記第2のアクティビティとのリンク情報を管理するサクセッサ・リストとを生成することを前記コンピュータに指示するプログラムコード手段と、

(c) 前記第2のアクティビティにおいて、前記モデレータ・テンプレートとのリンク情報を管理するプレデセッサ・リストを生成することを前記コンピュータに指示するプログラムコード手段と、
10 を含む記憶媒体。

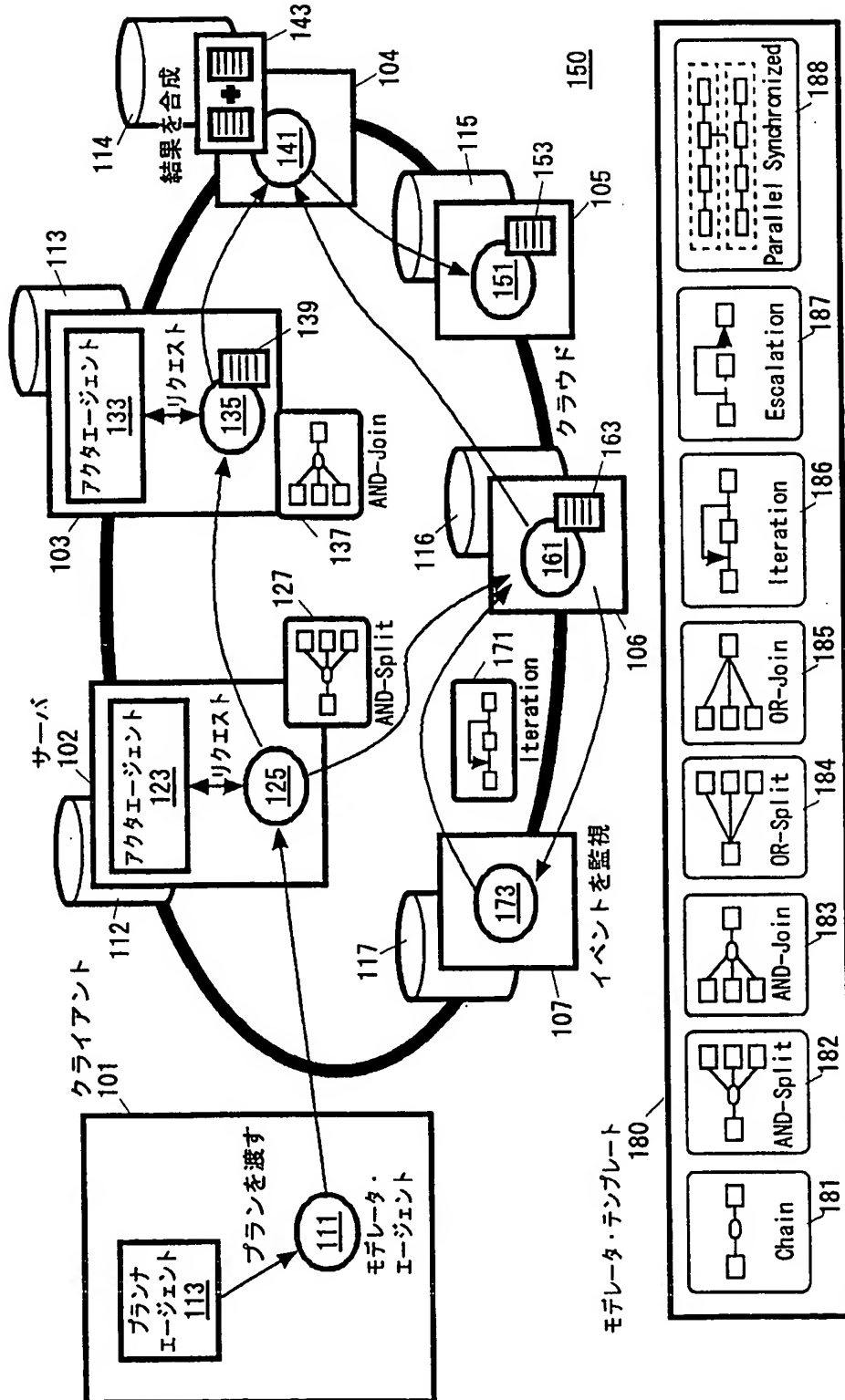


FIG. 1

2/29

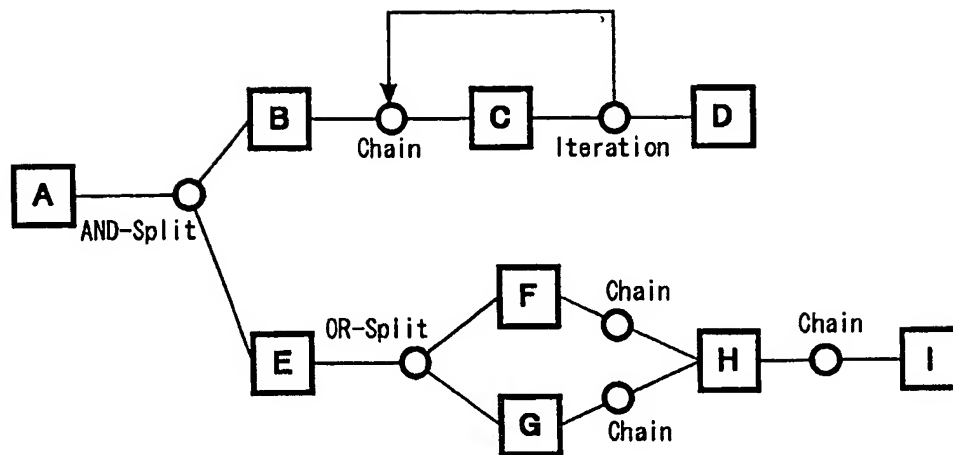


FIG. 2

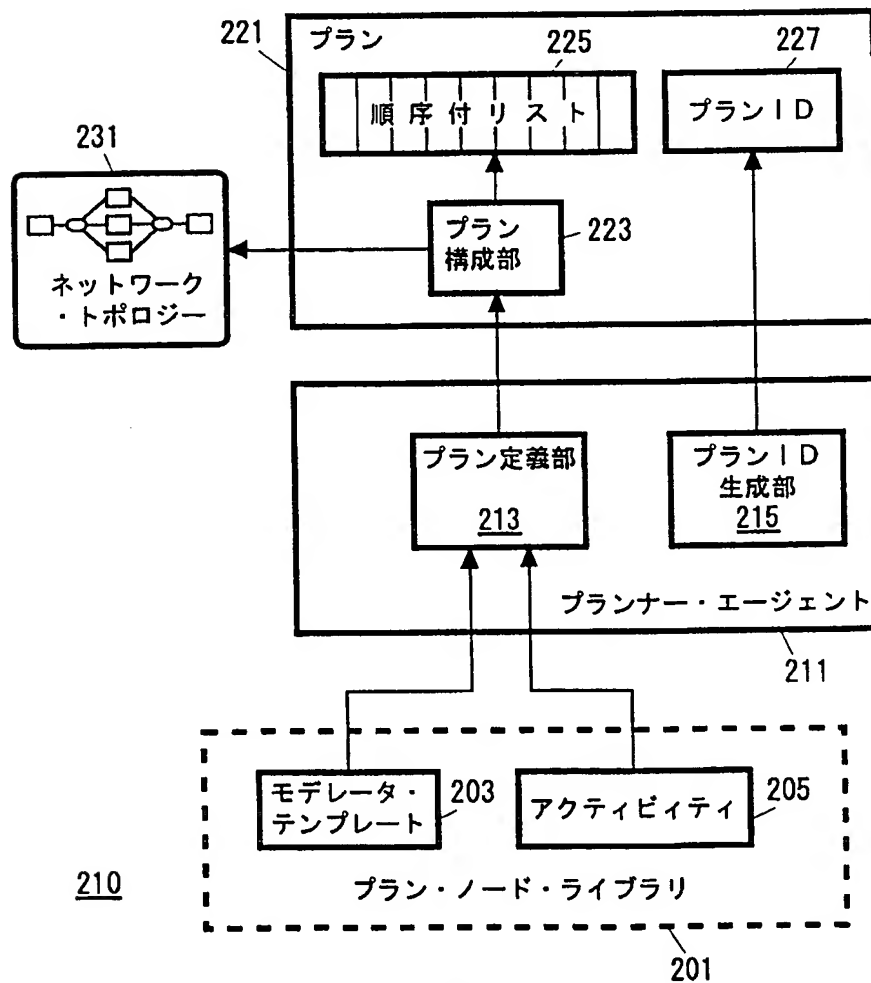


FIG. 4

3/29

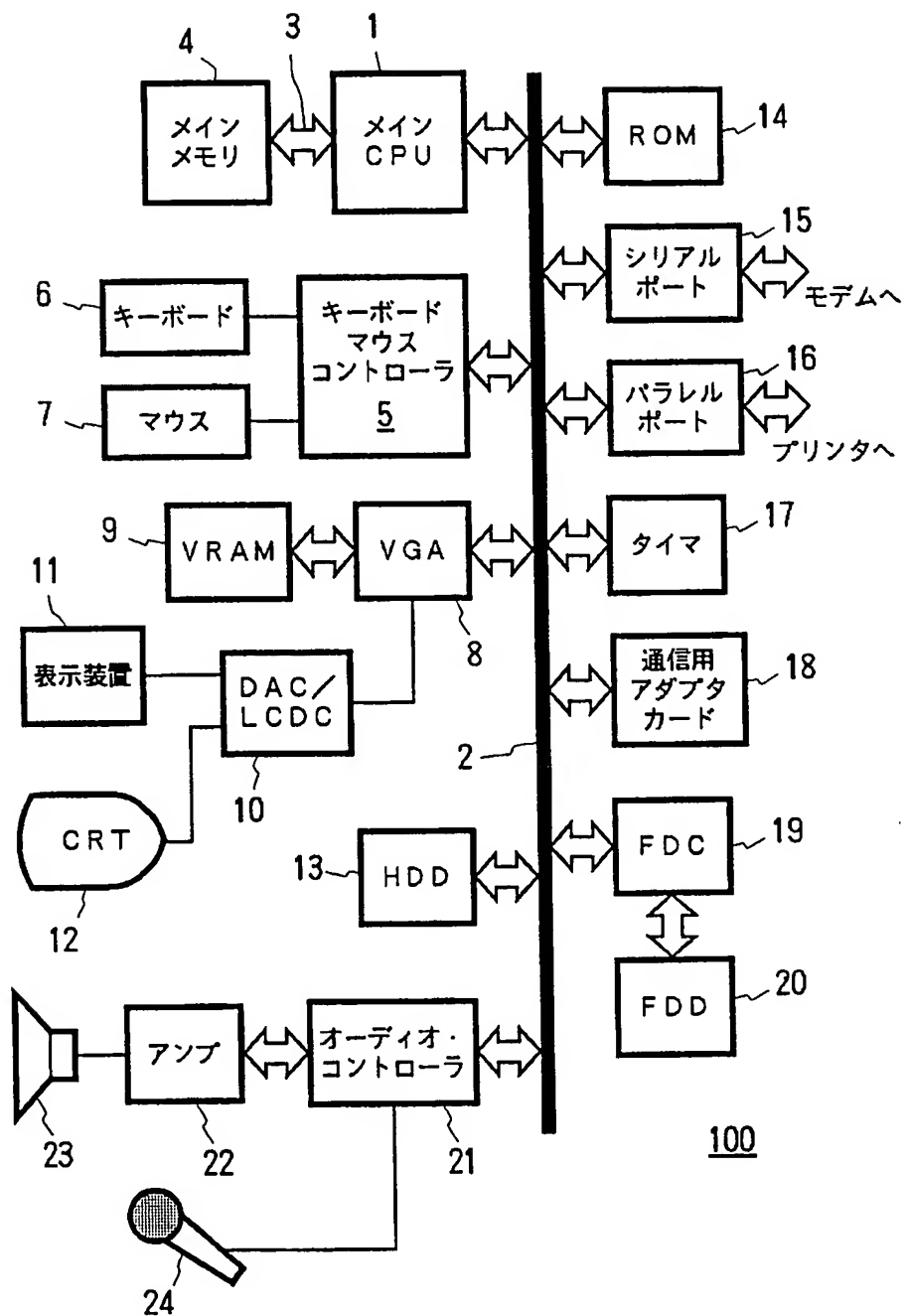


FIG. 3

4/29

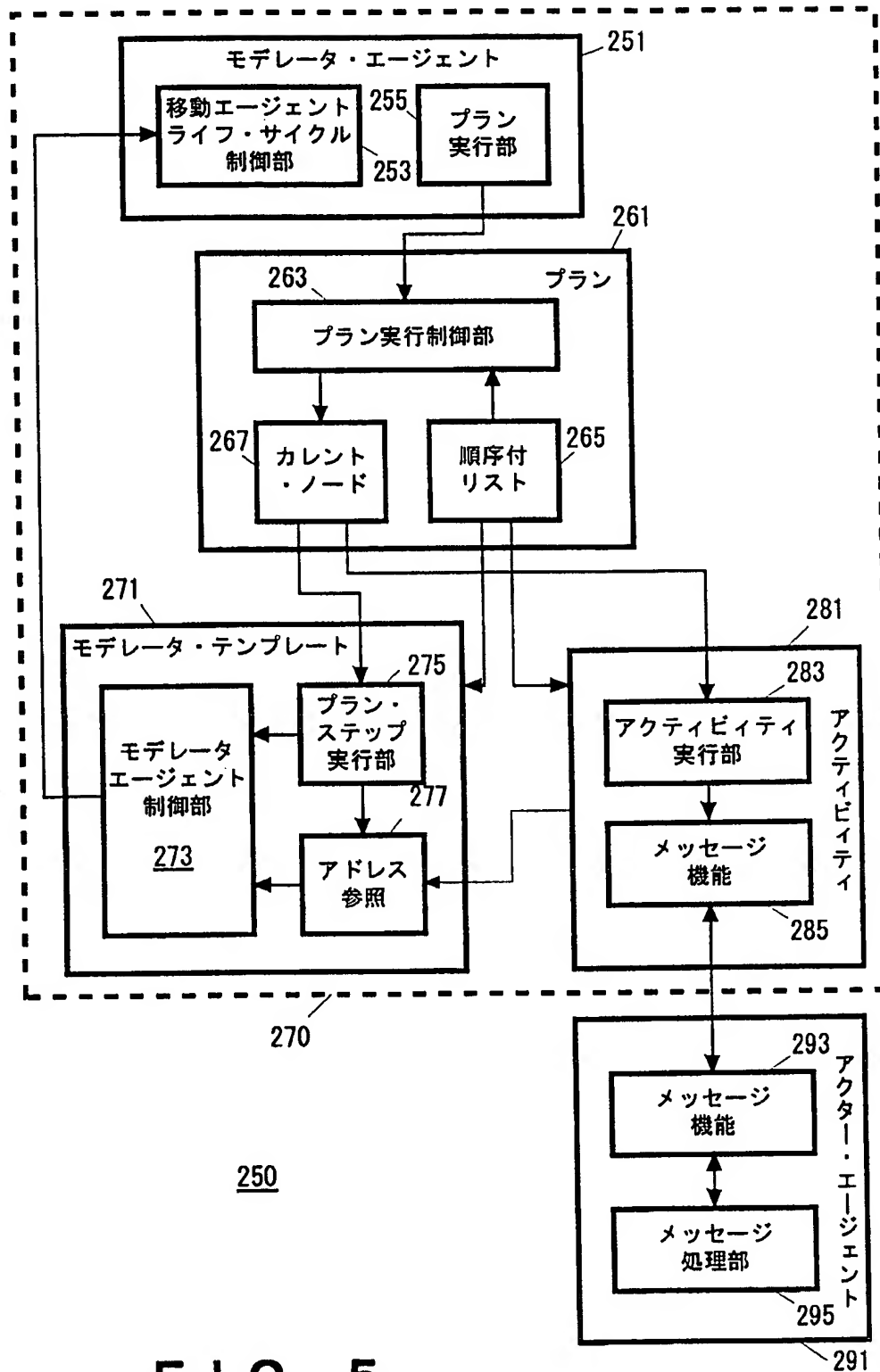


FIG. 5

5/29

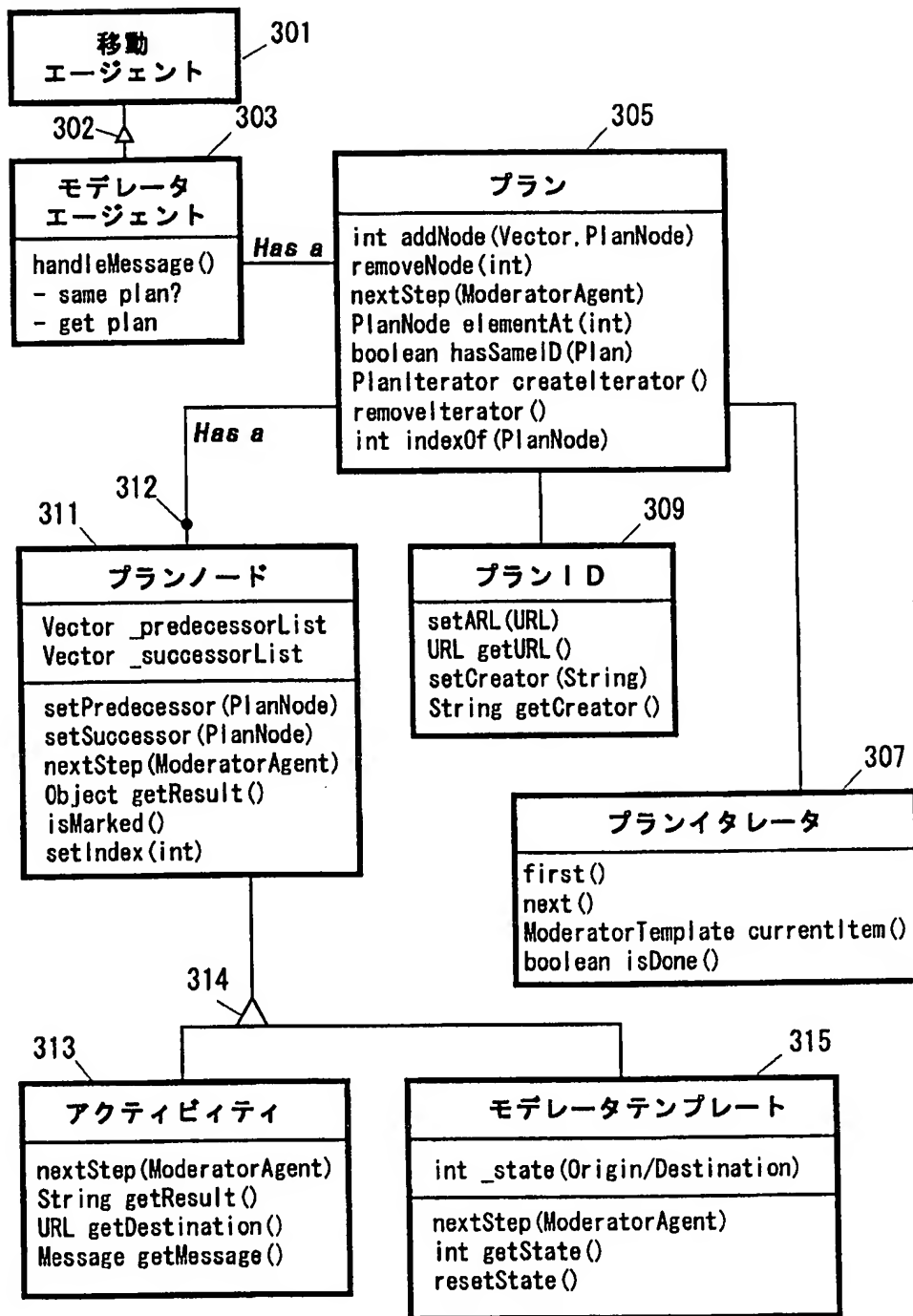


FIG. 6

6/29

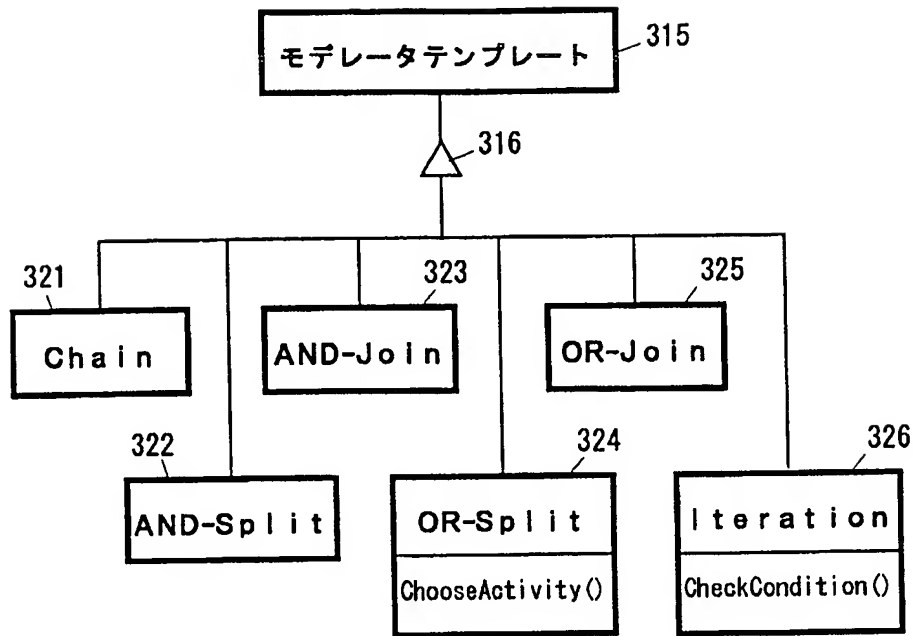


FIG. 7

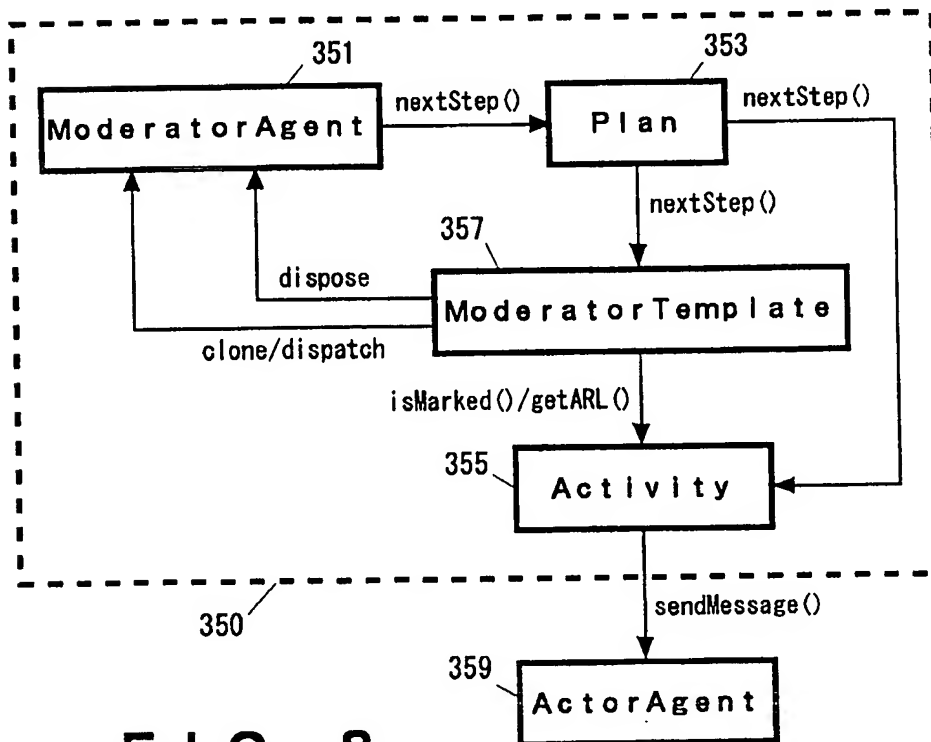


FIG. 8

7/29

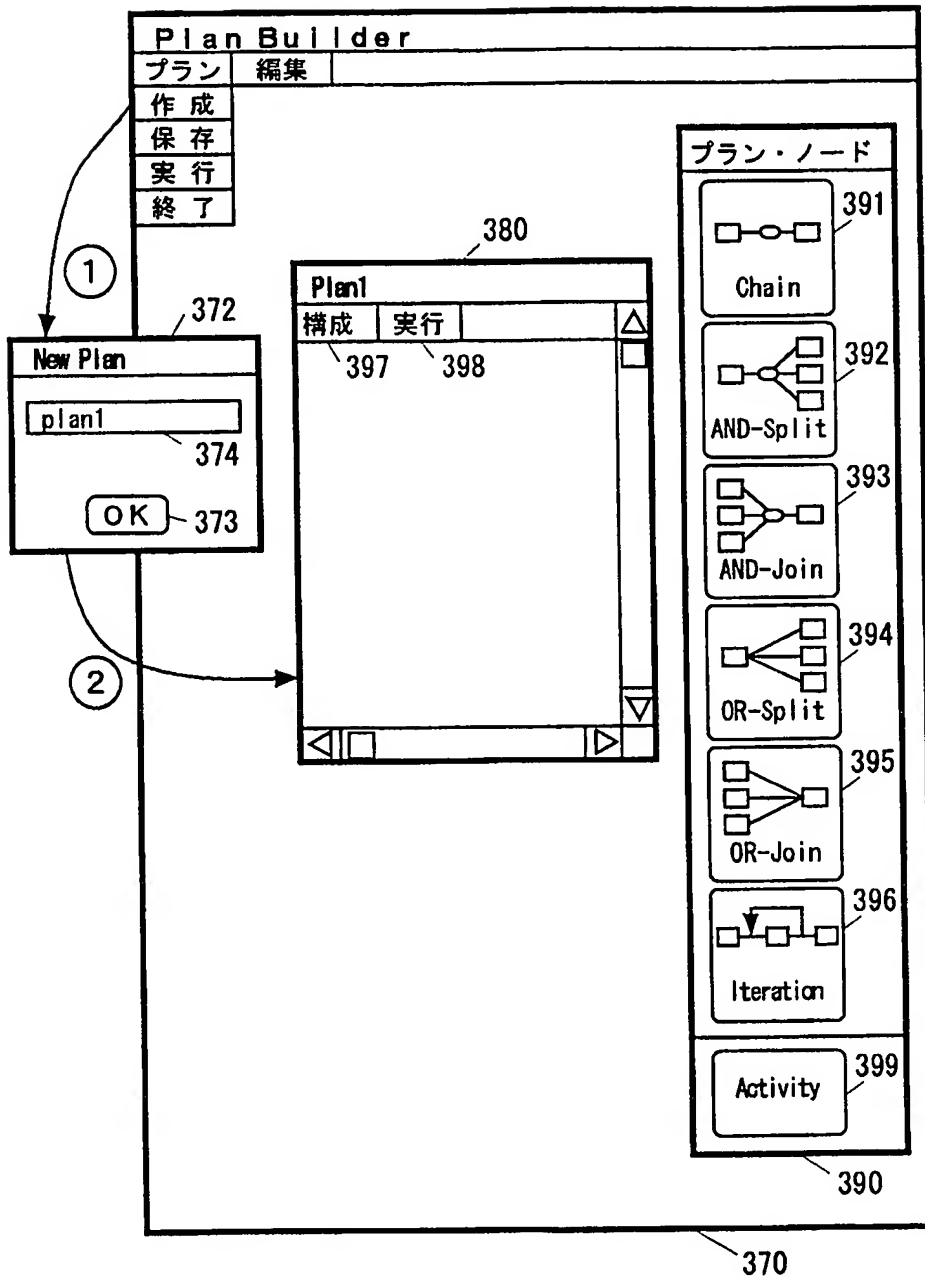


FIG. 9

8/29

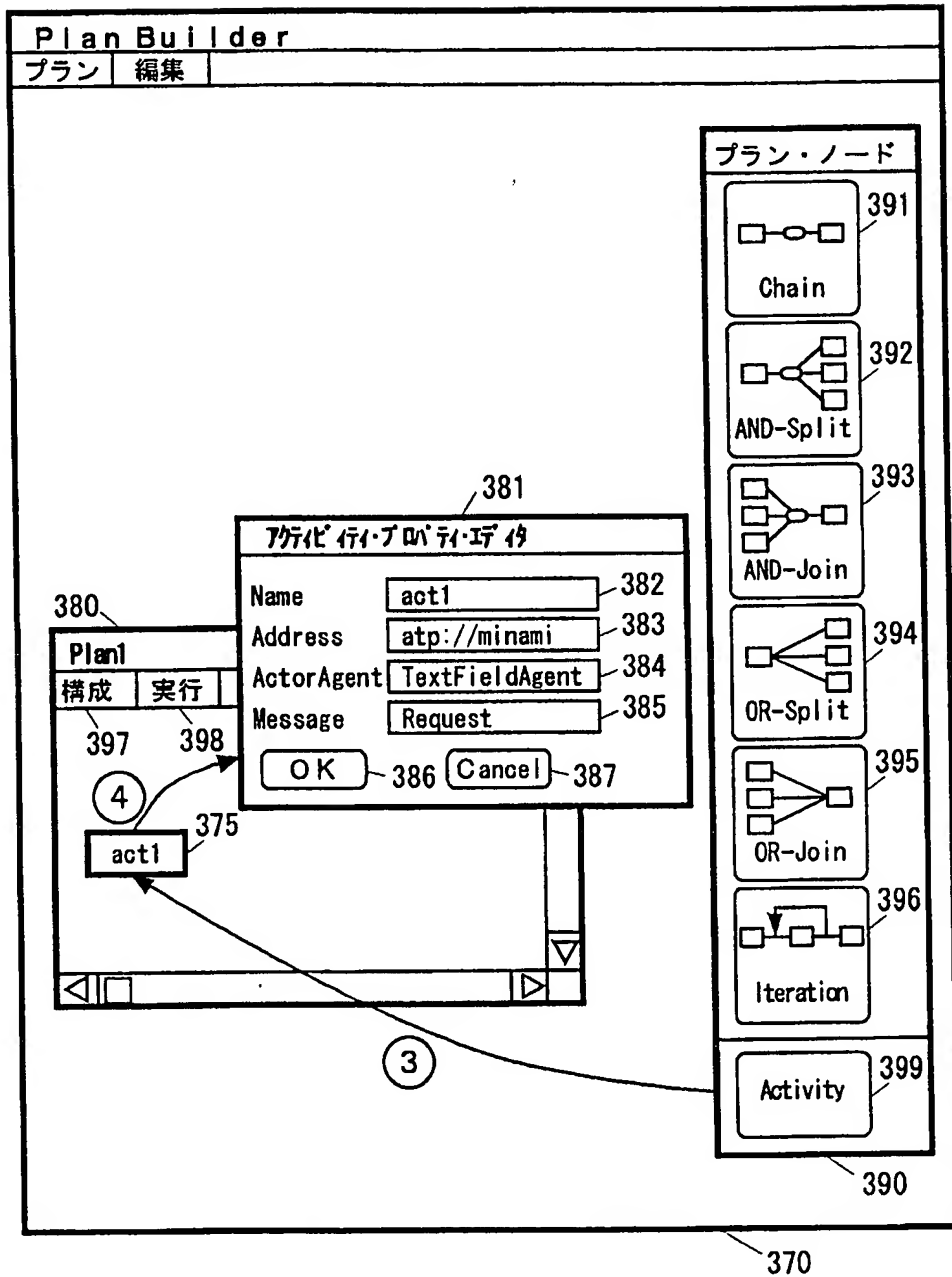


FIG. 10

9/29

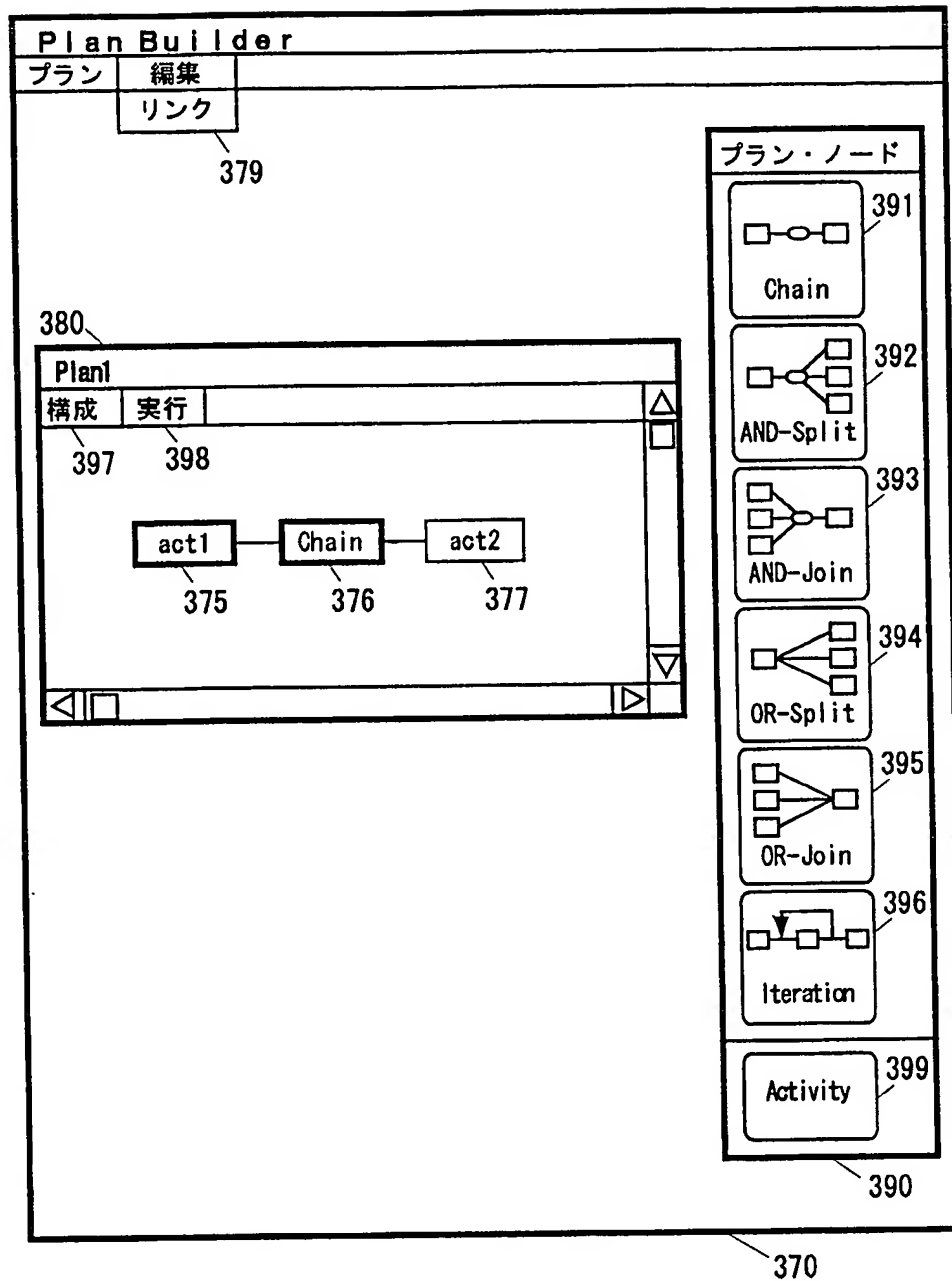


FIG. 11

10/29

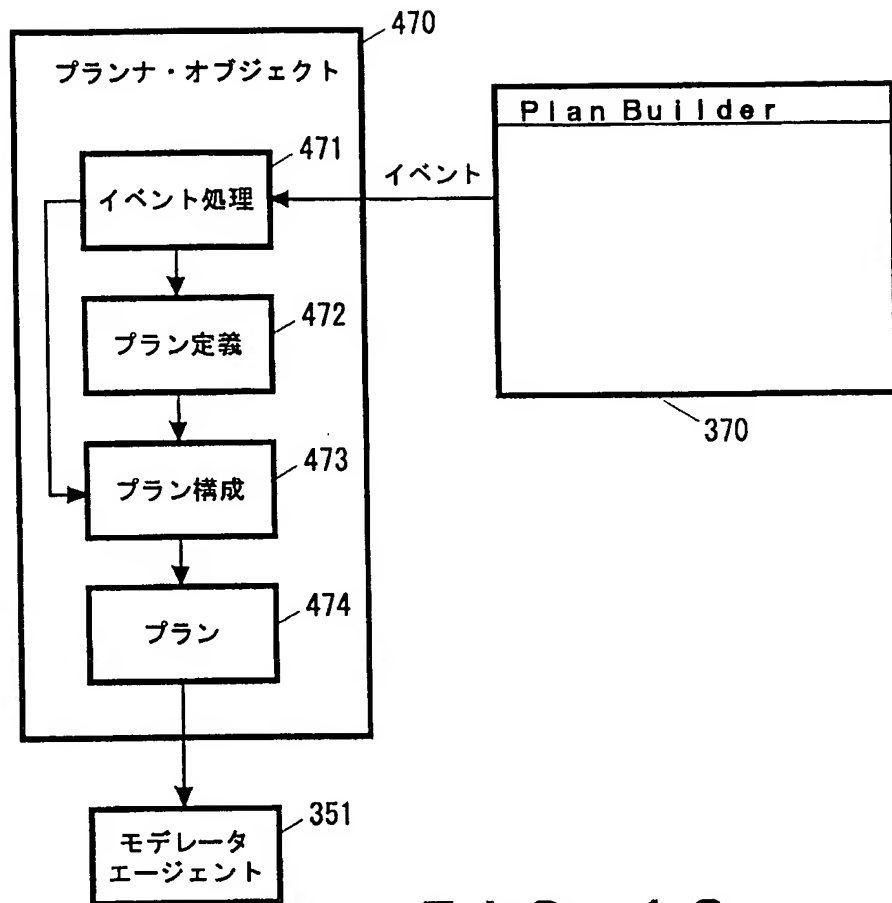


FIG. 12

11/29

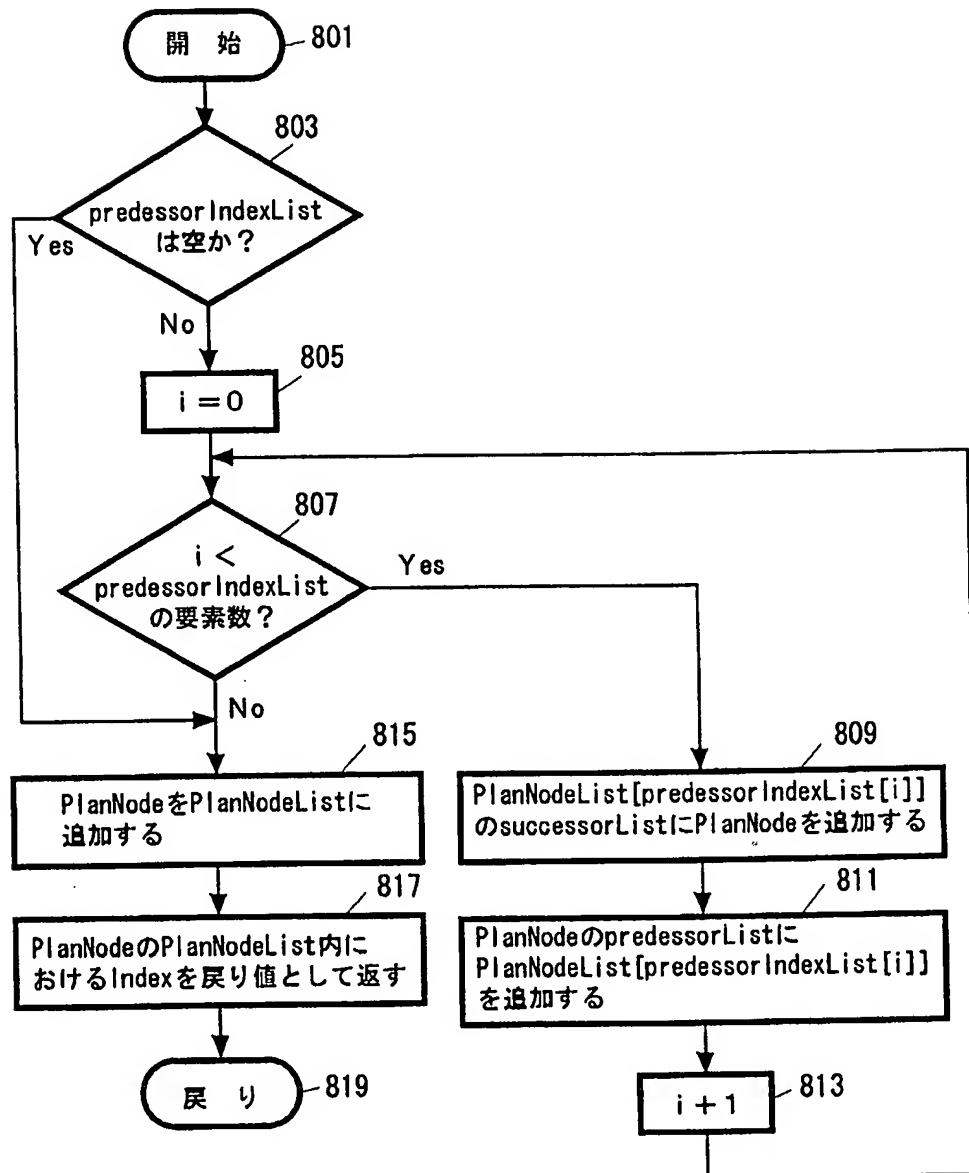


FIG. 13

12/29

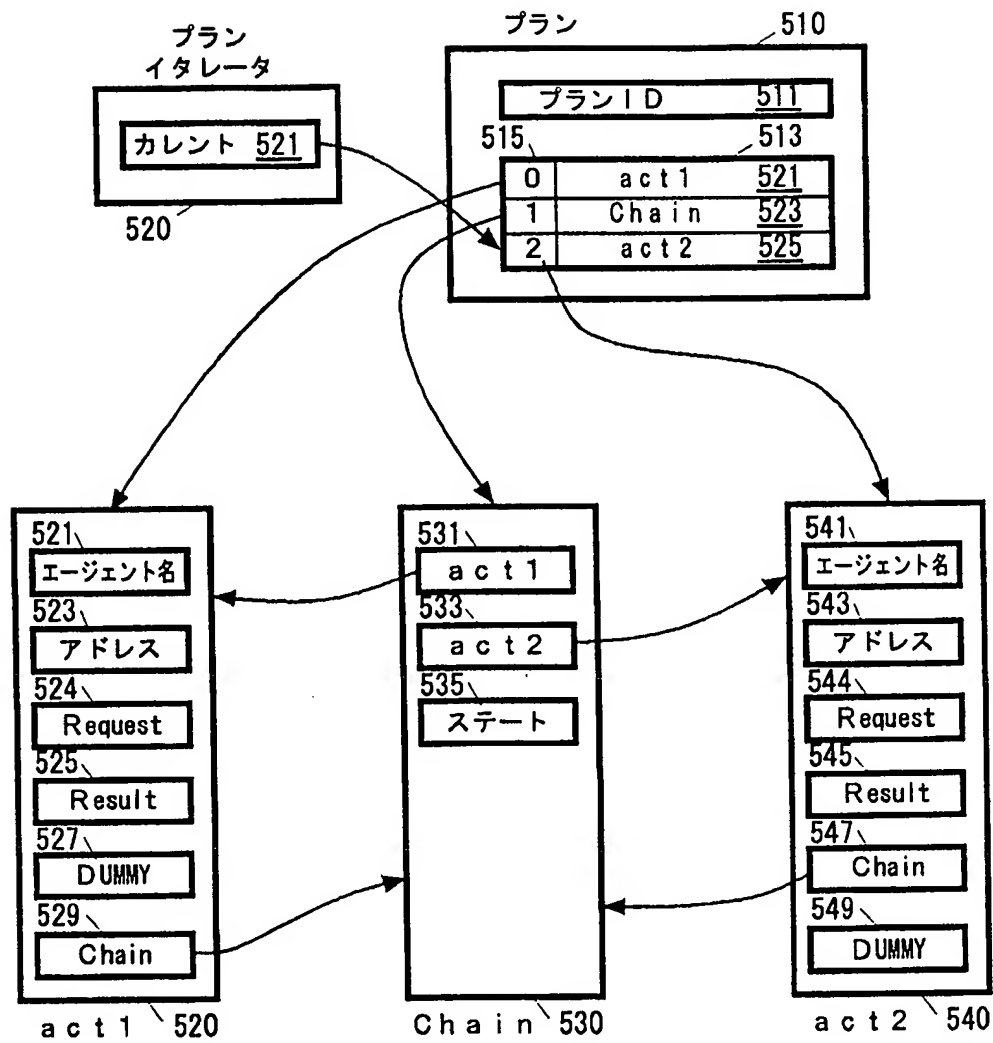


FIG. 14

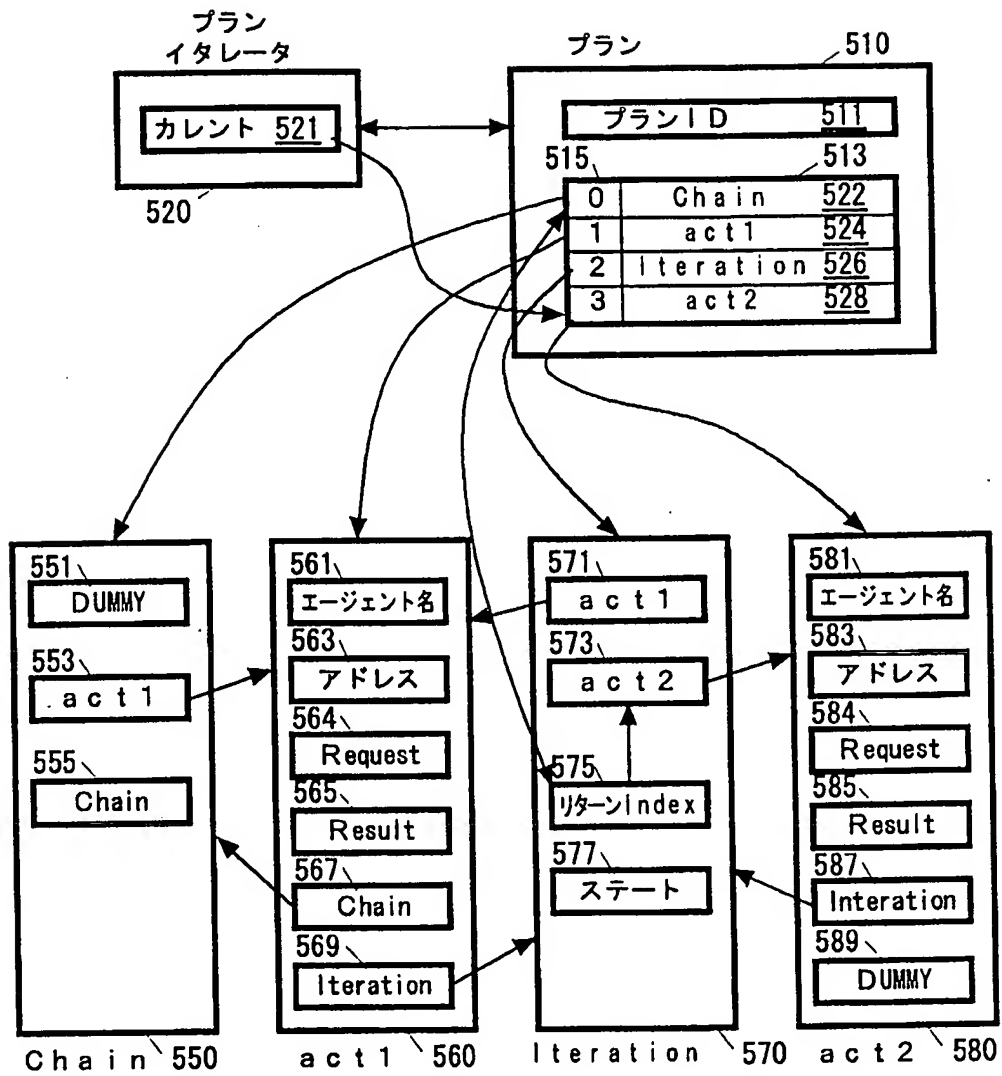


FIG. 15

14/29

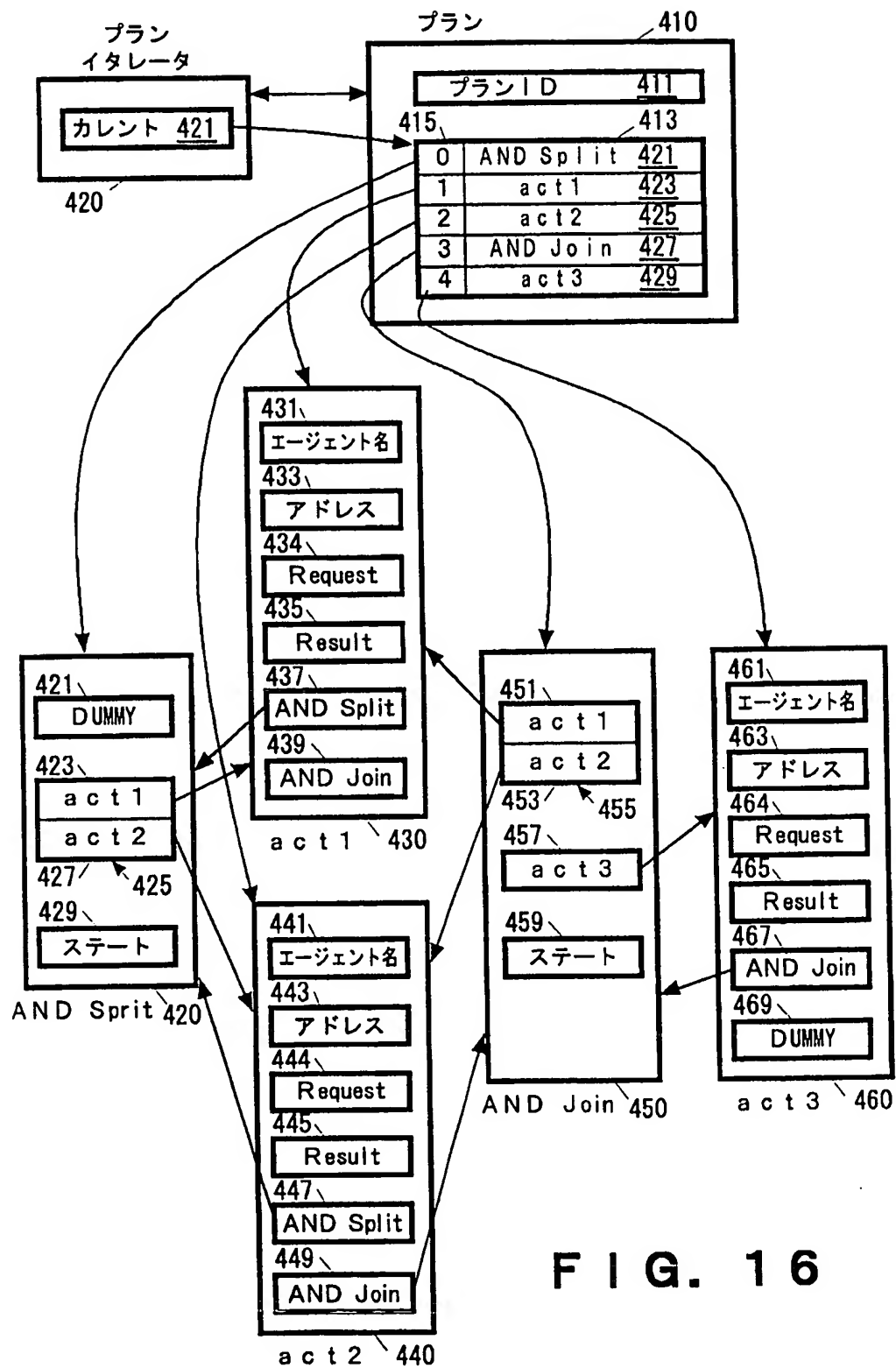


FIG. 16

15/29

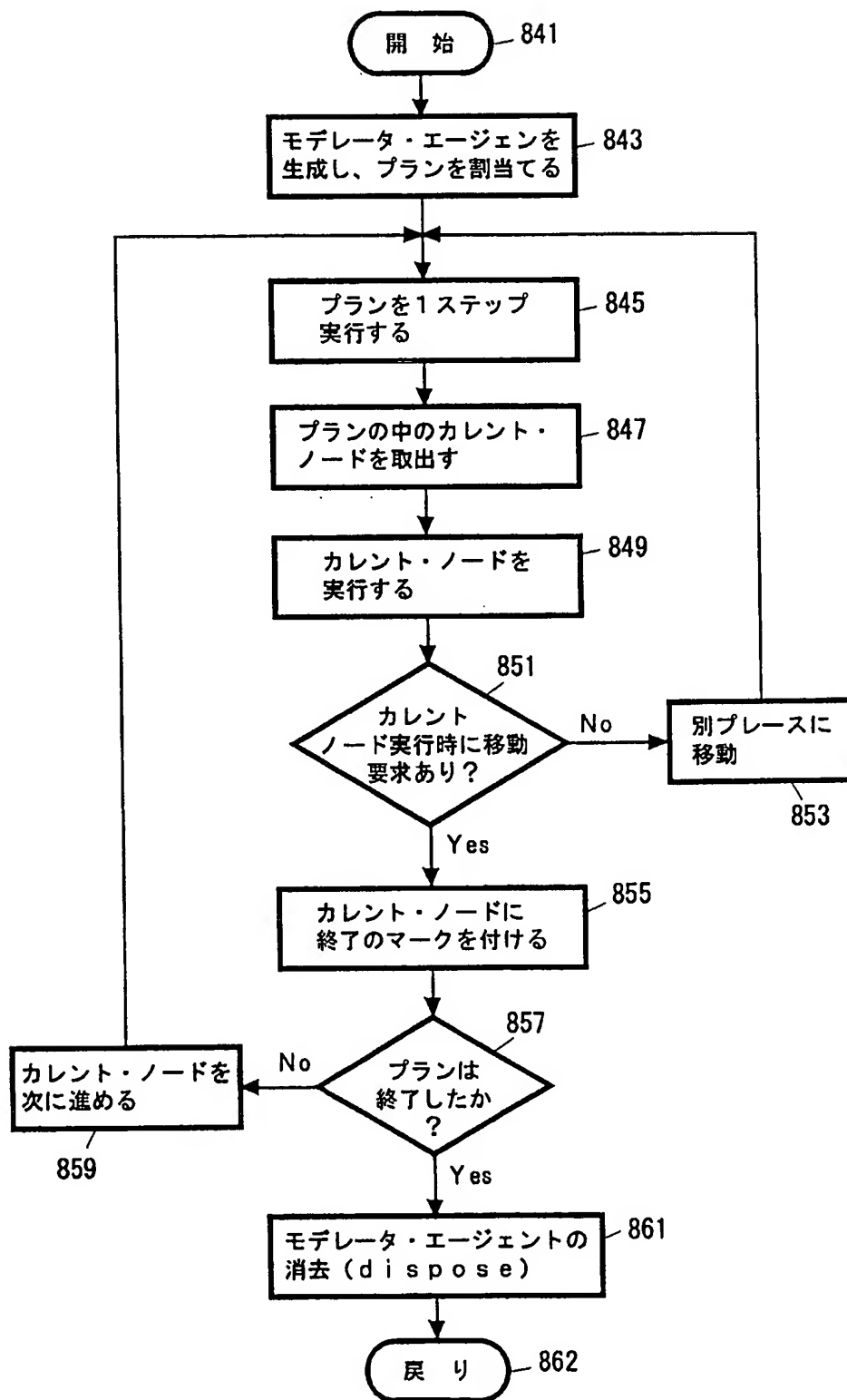


FIG. 17

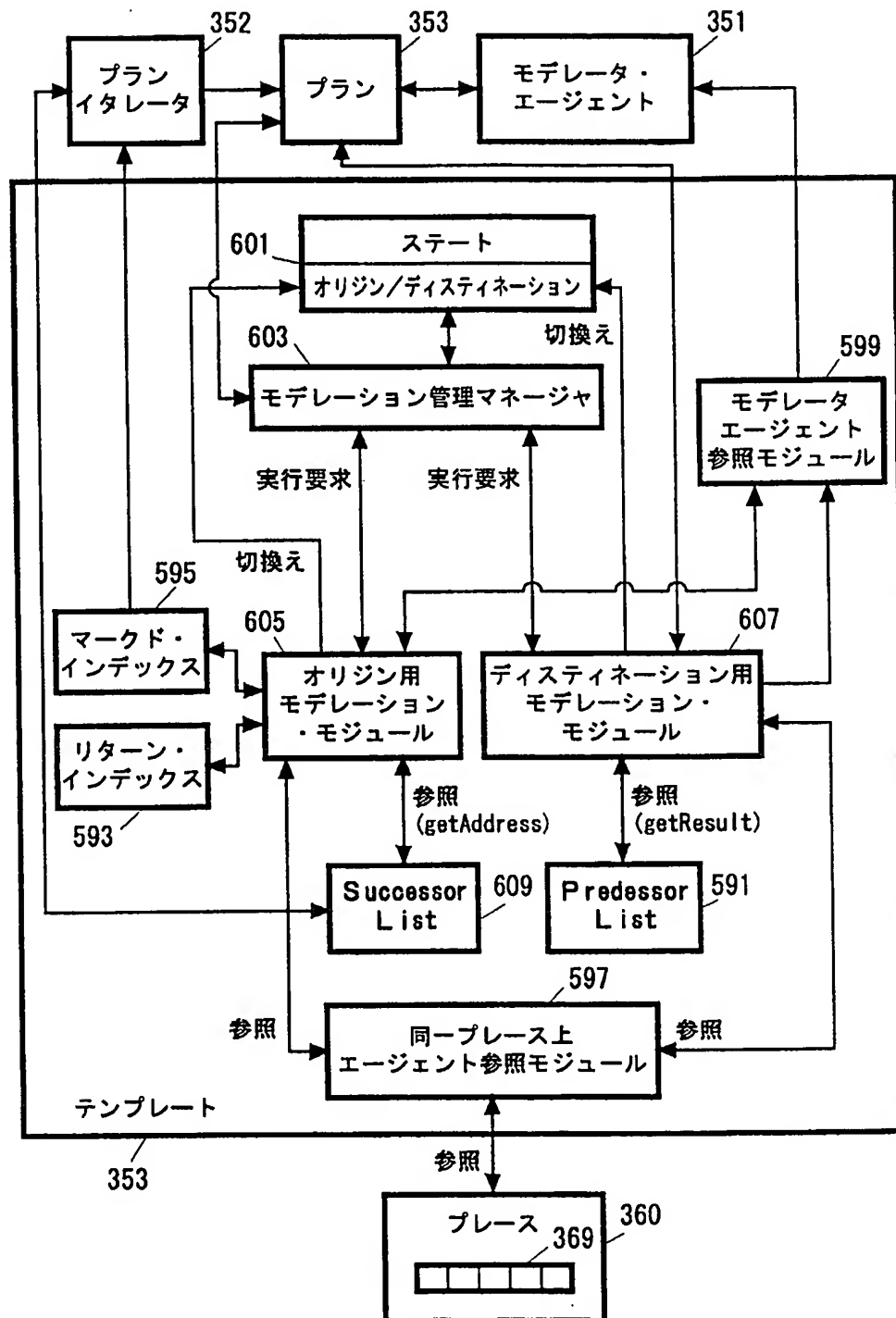


FIG. 18

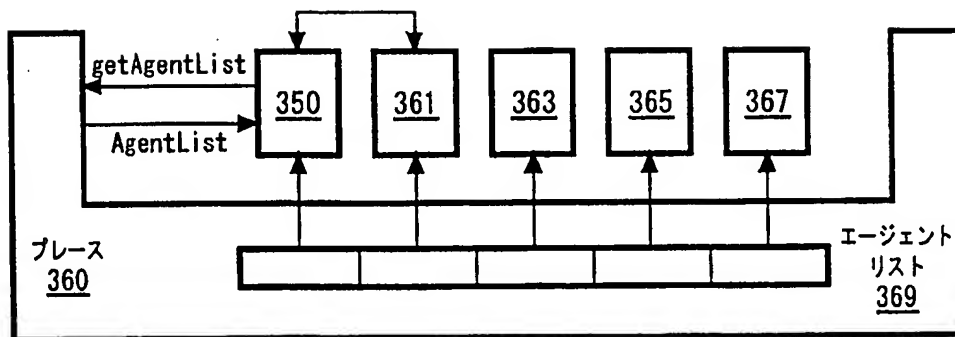


FIG. 19

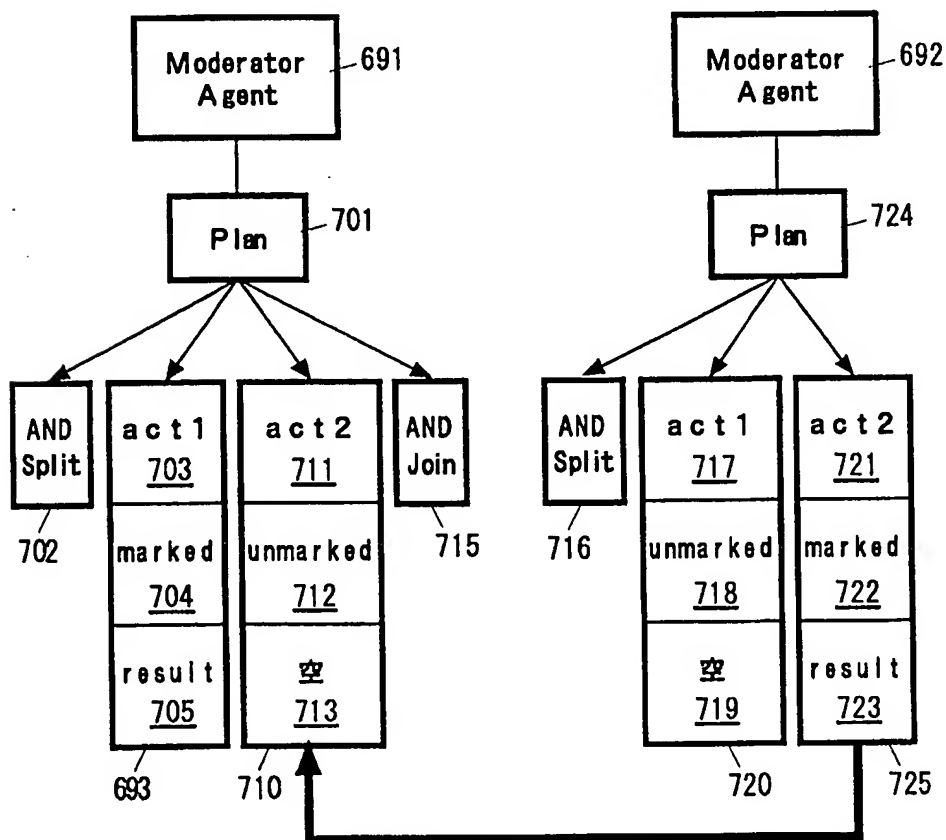


FIG. 30

18/29

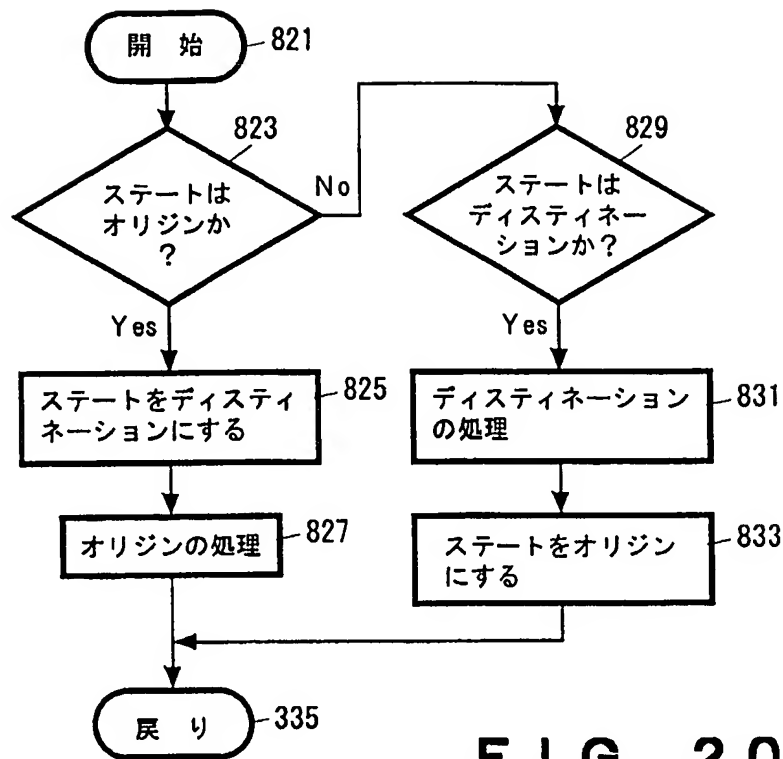


FIG. 20

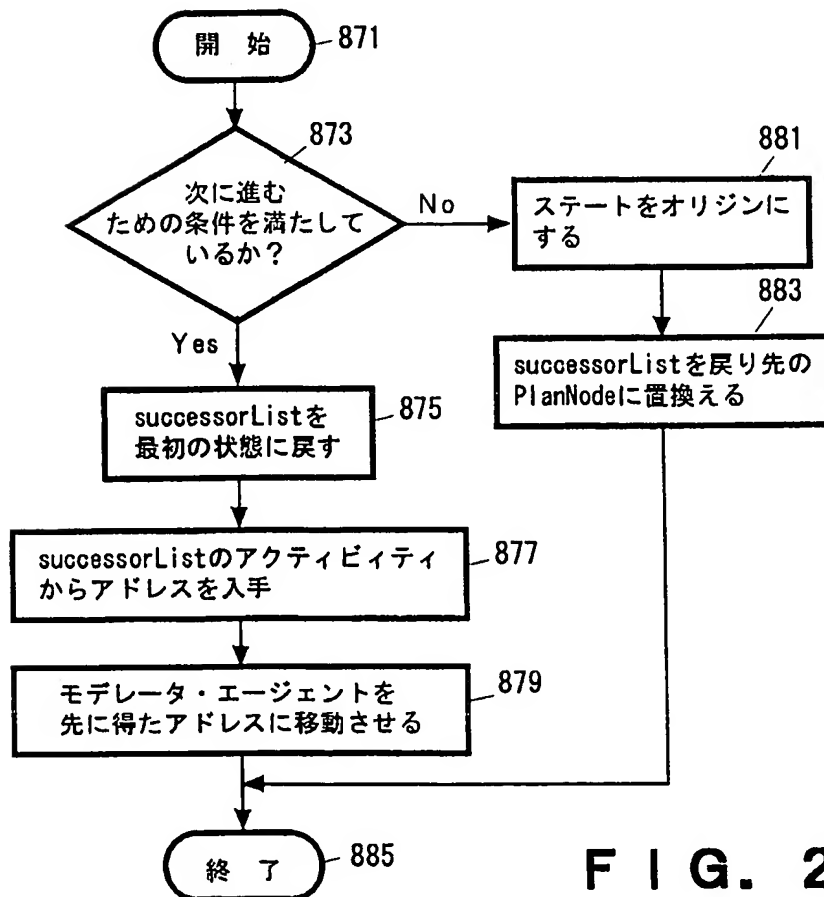


FIG. 23

19/29

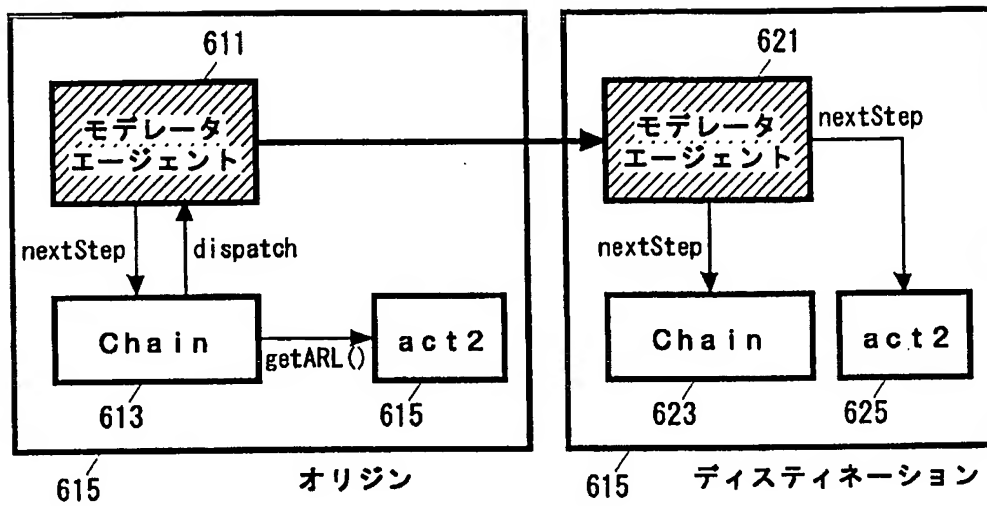


FIG. 21

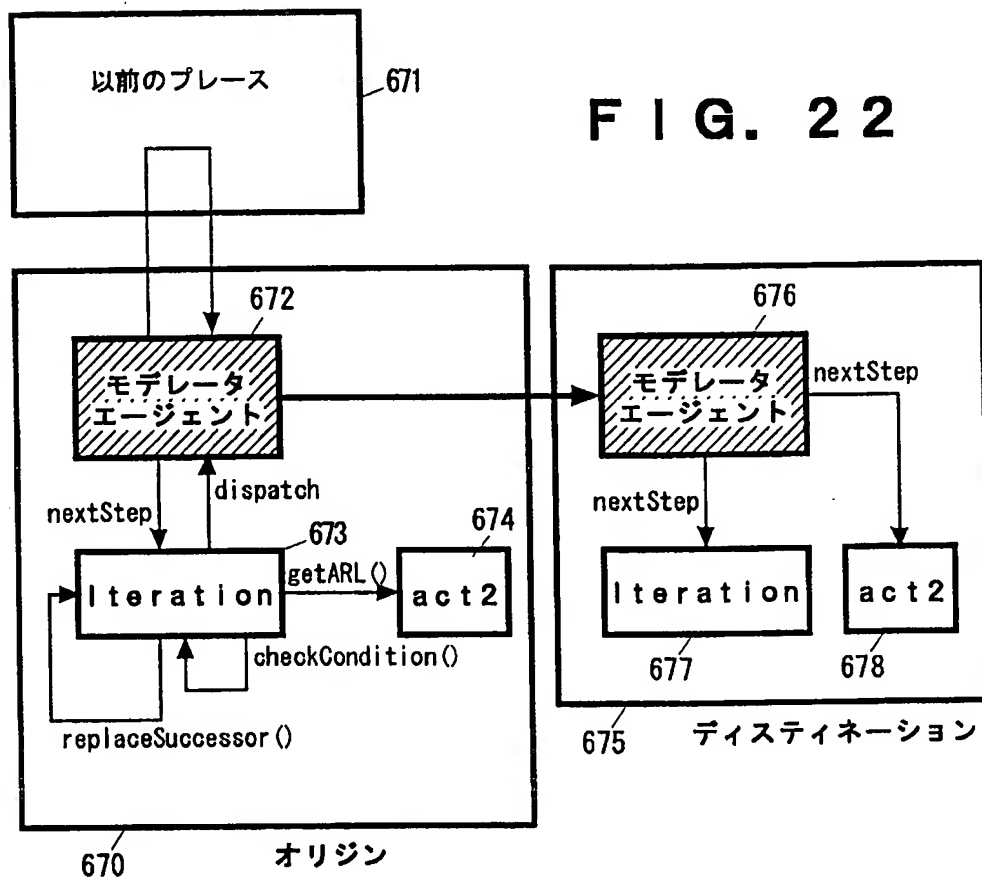


FIG. 22

20/29

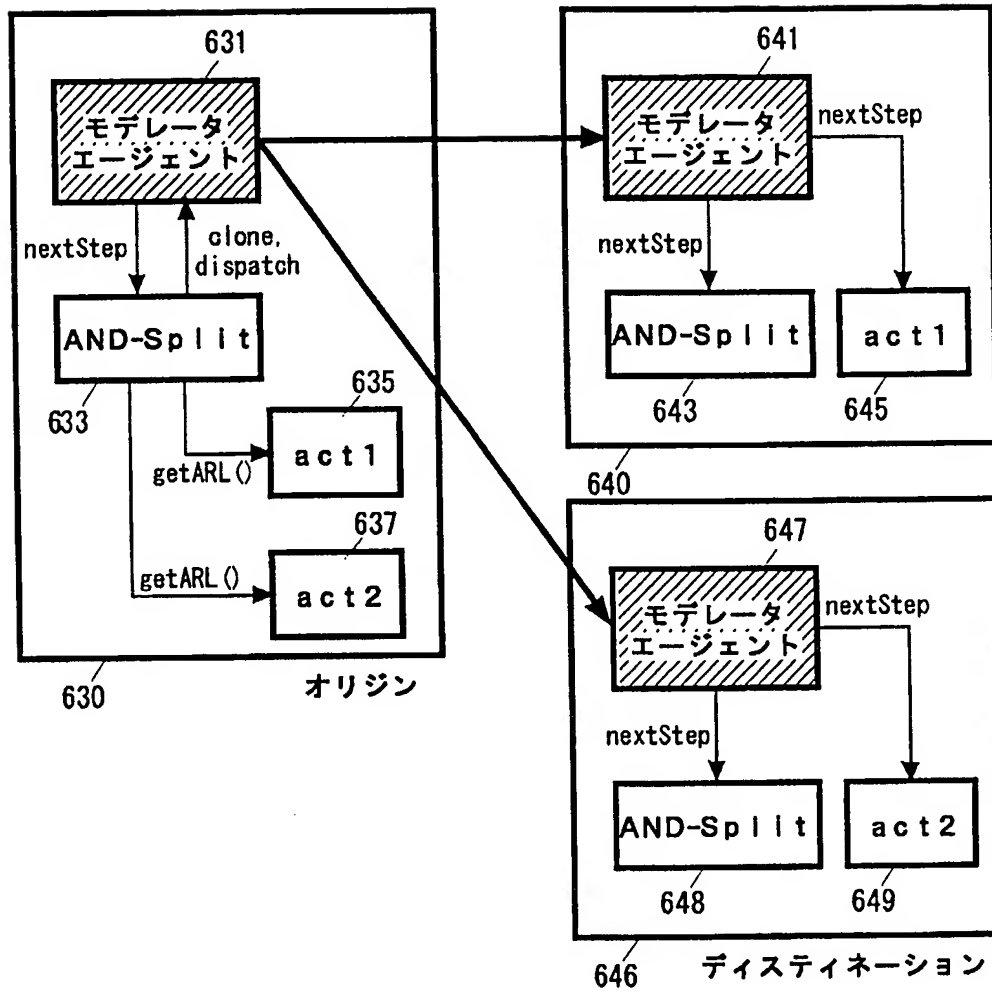


FIG. 24

21/29

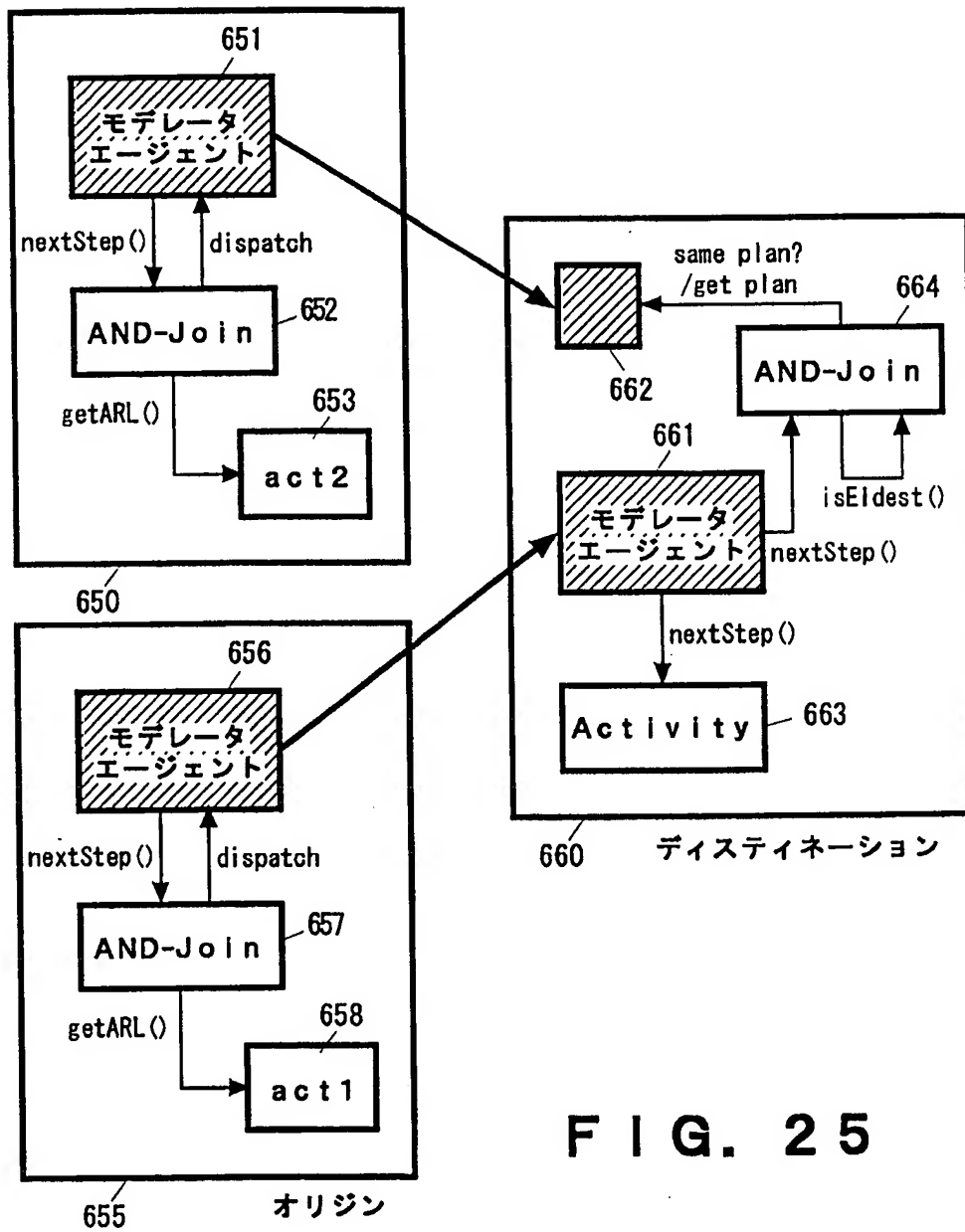


FIG. 25

22/29

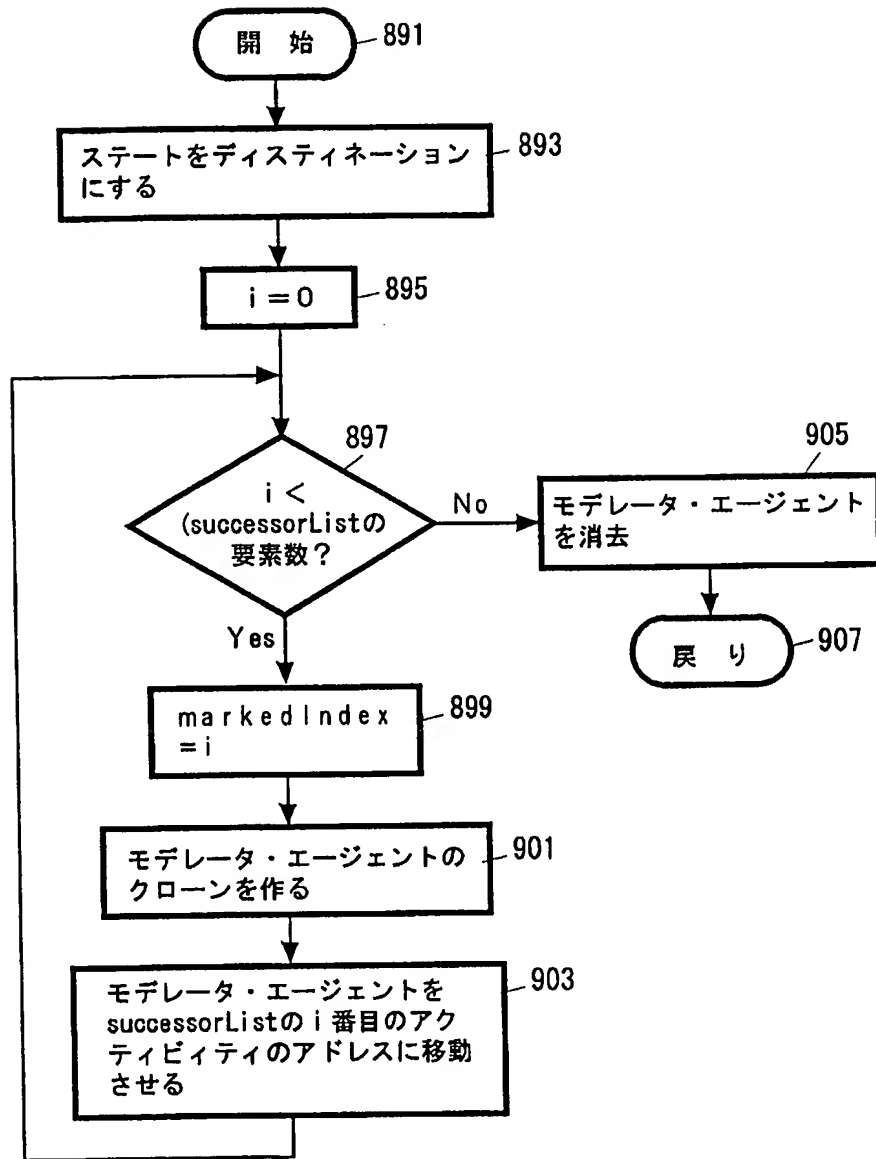


FIG. 26

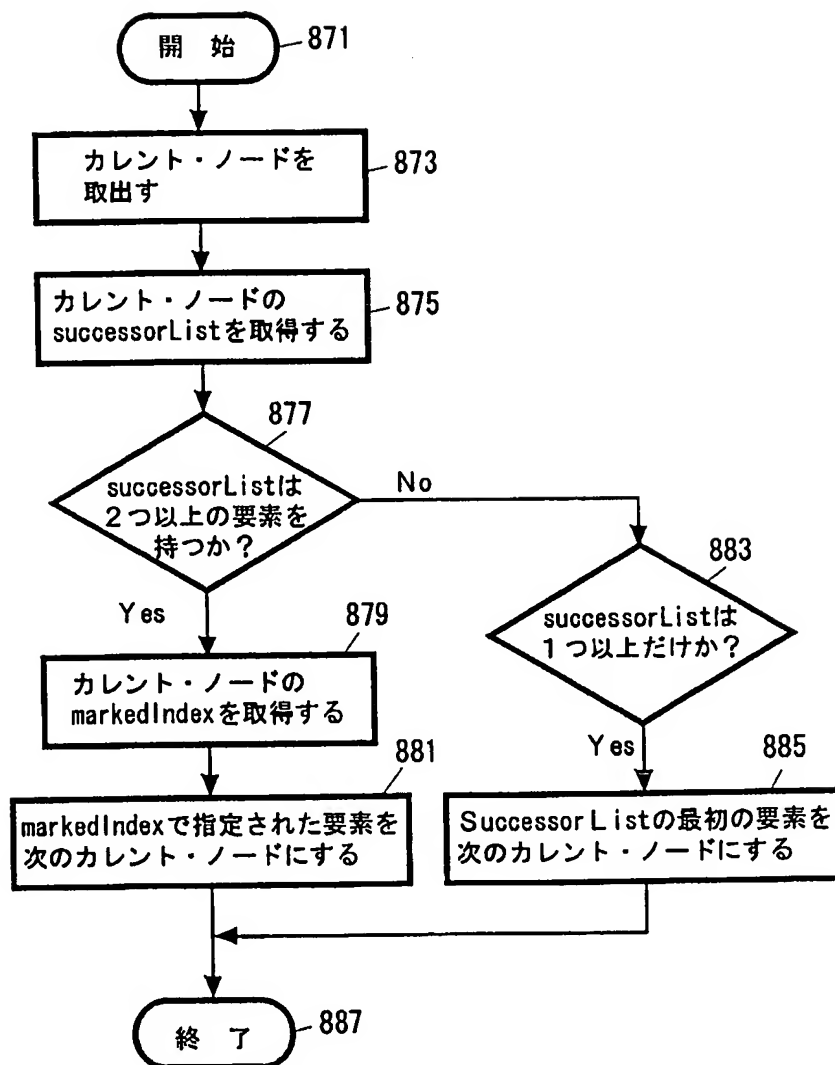


FIG. 27

24/29

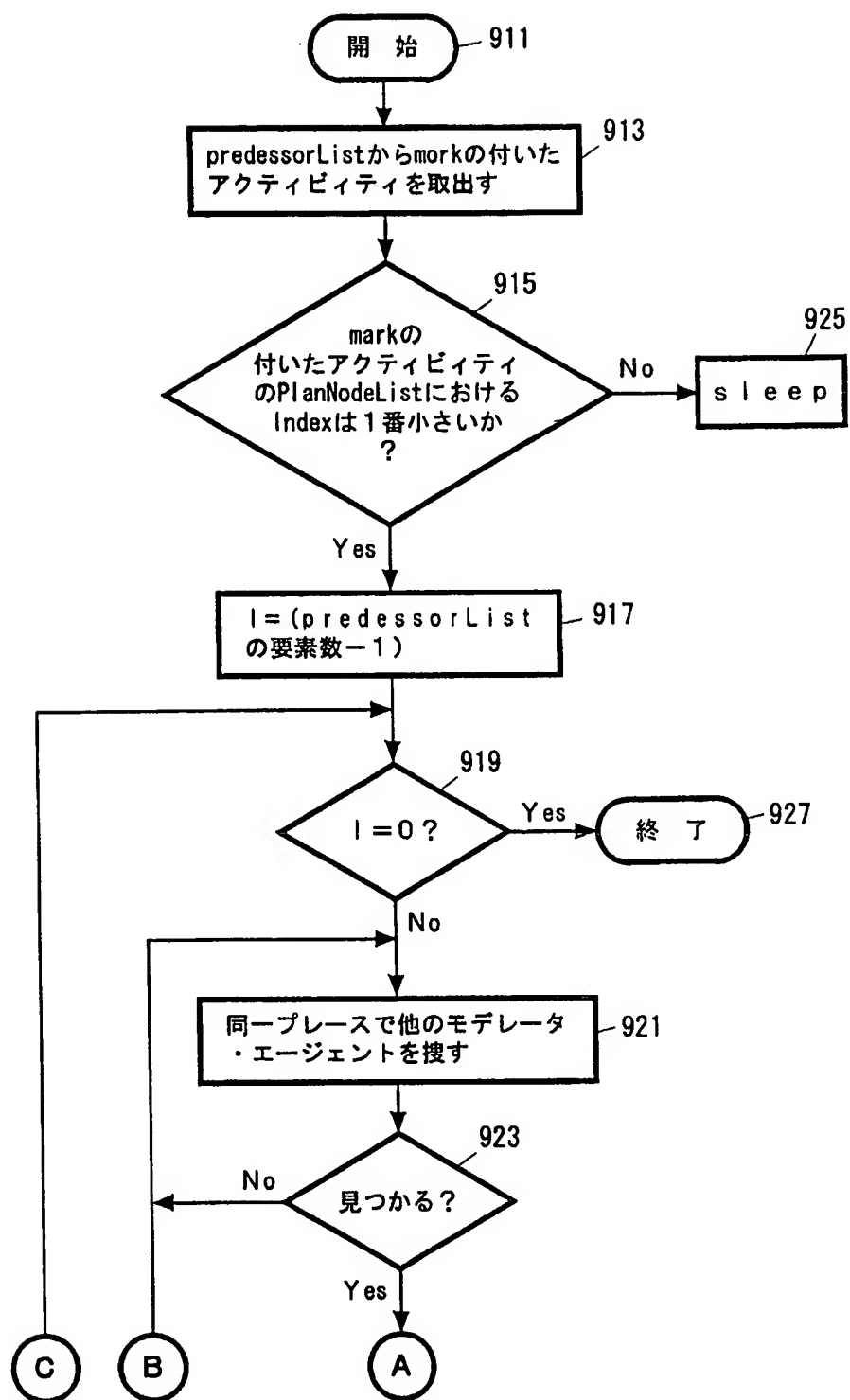


FIG. 28

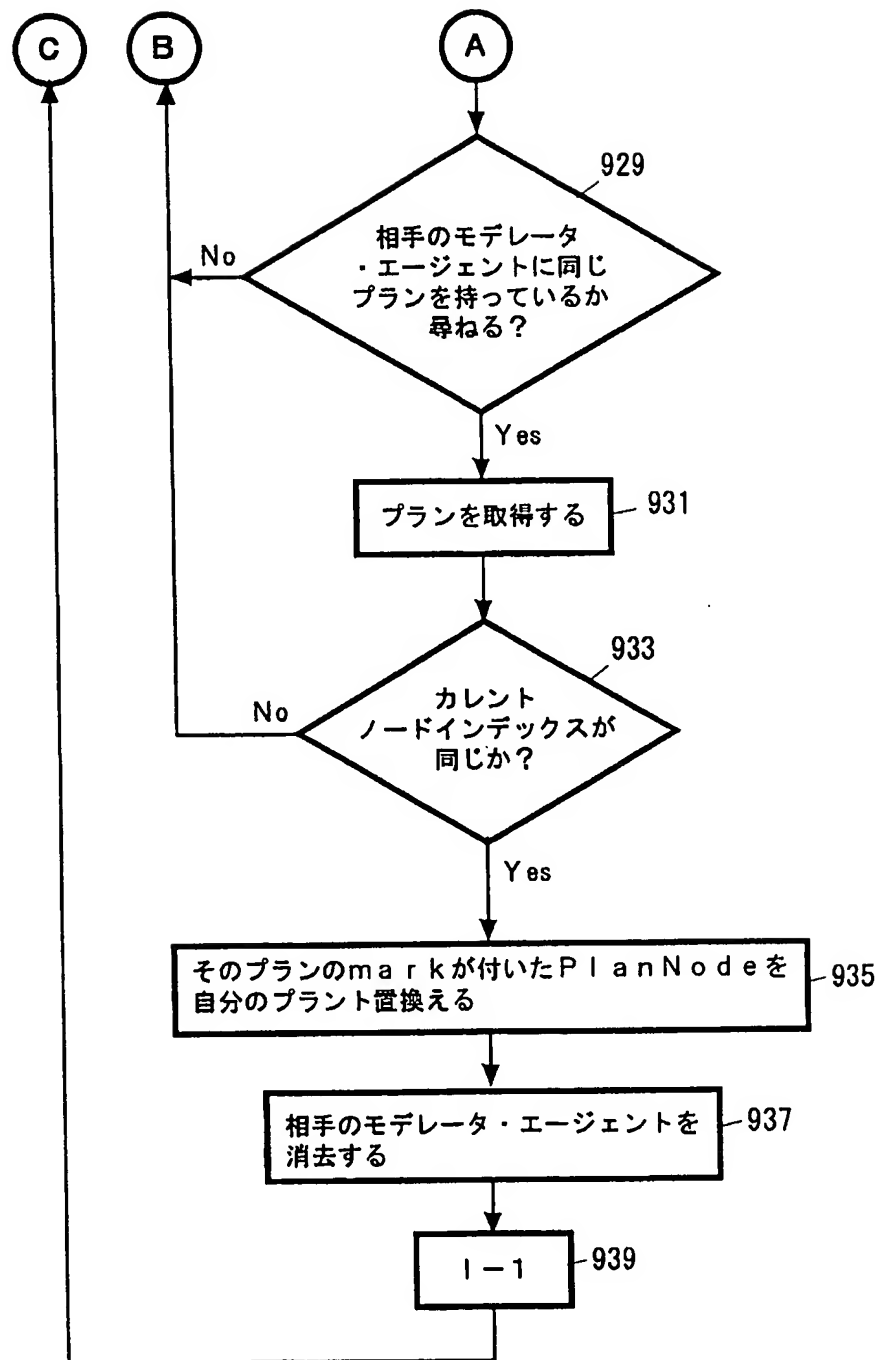
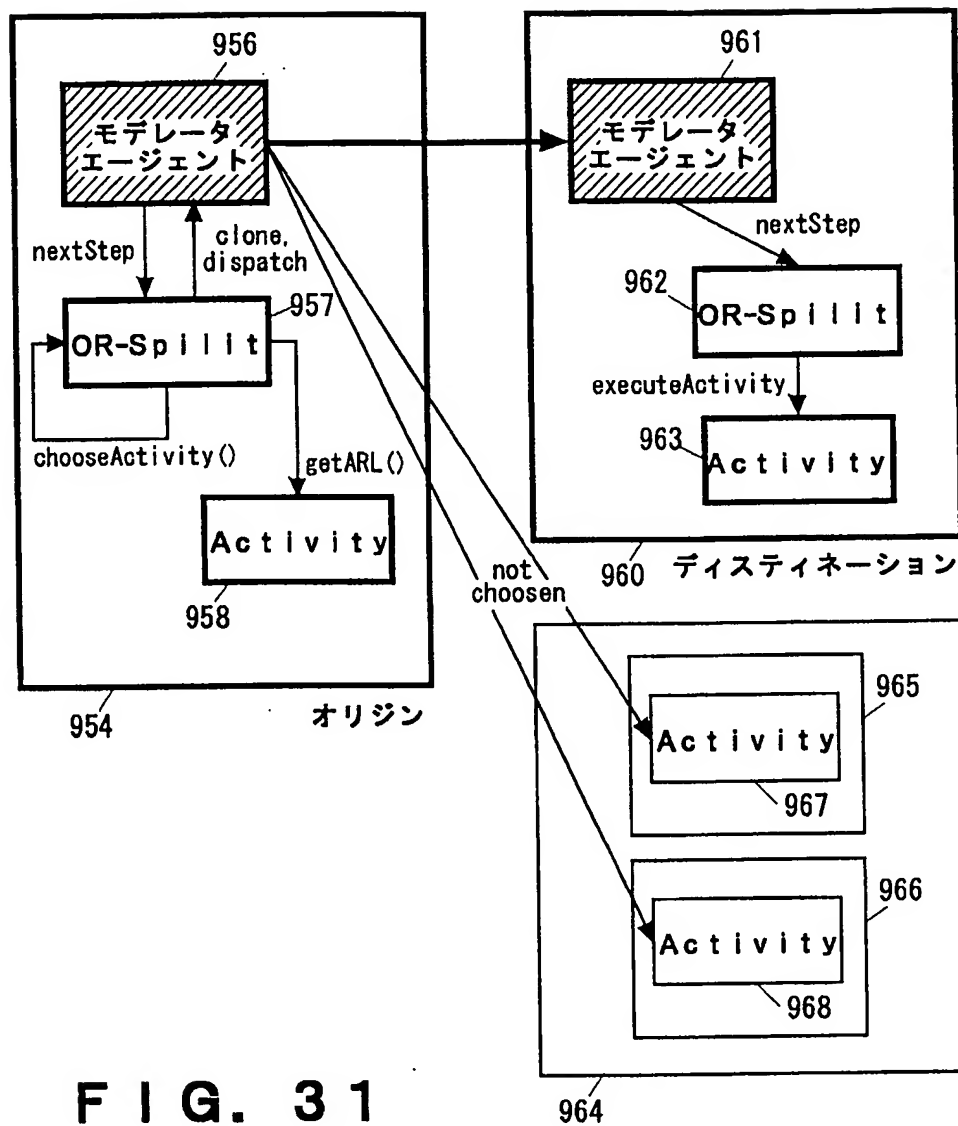


FIG. 29



27/29

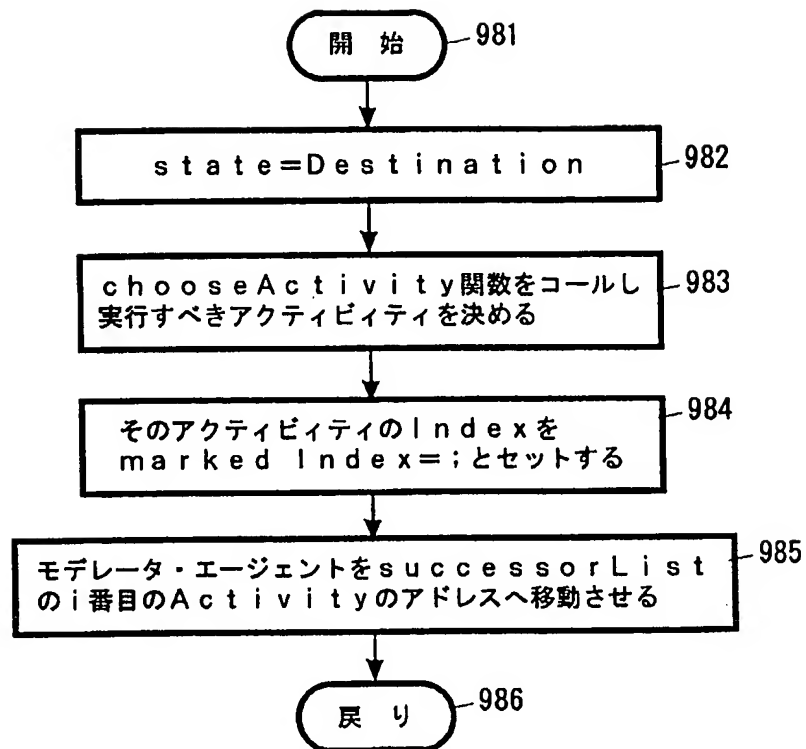


FIG. 32

28/29

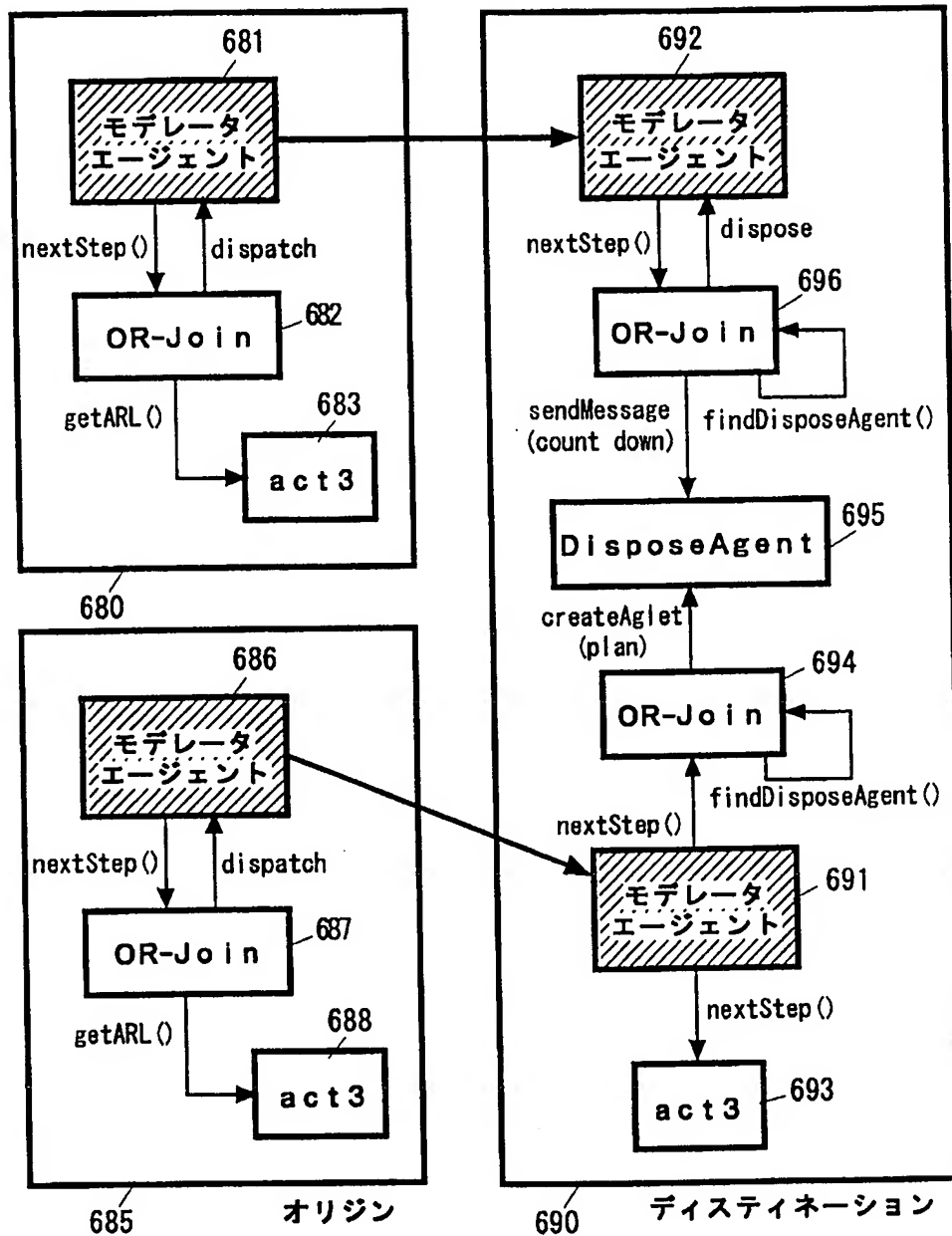


FIG. 33

29/29

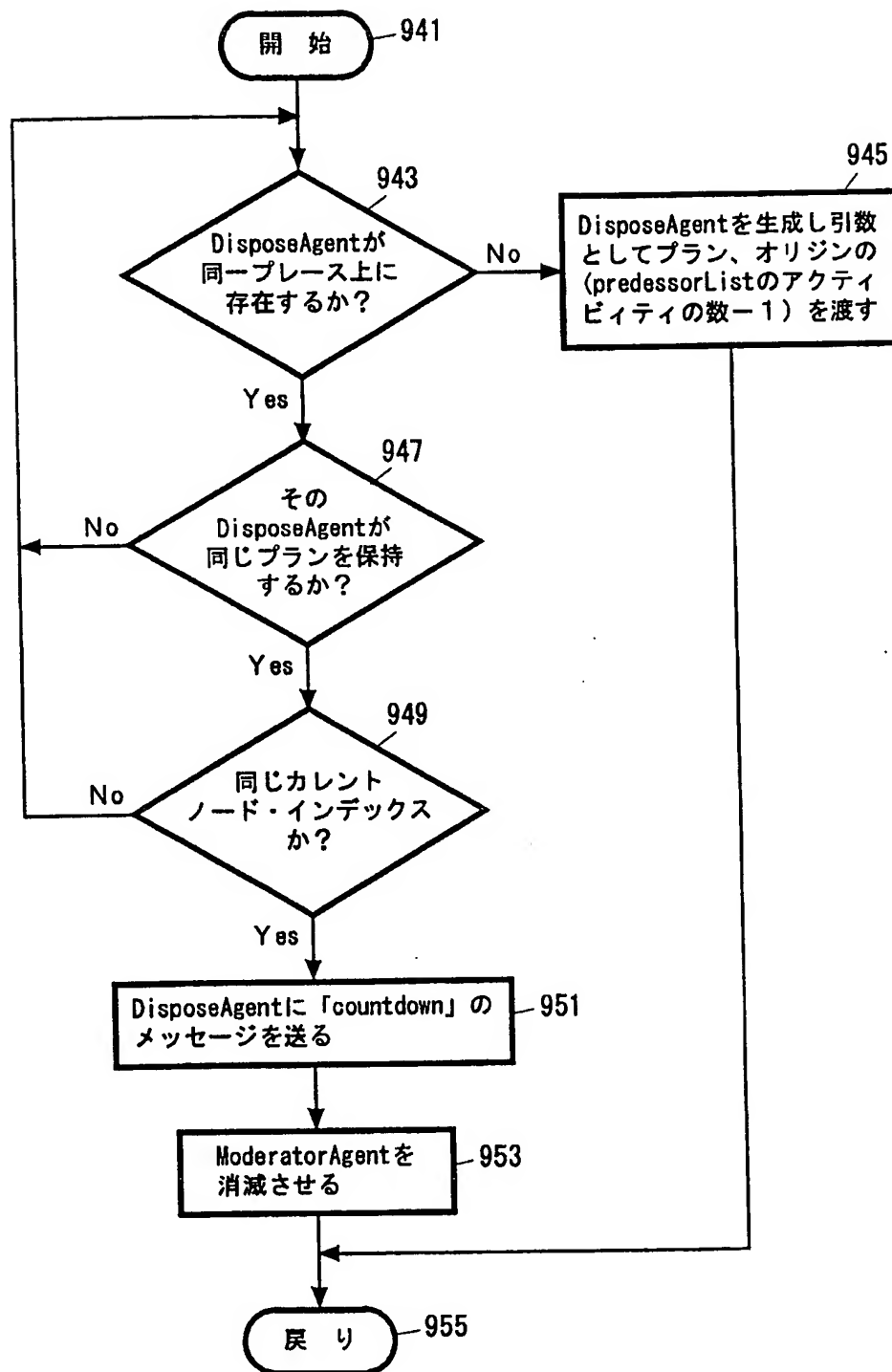


FIG. 34

INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP97/04864

A. CLASSIFICATION OF SUBJECT MATTER

Int.Cl⁶ G06F9/44

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

Int.Cl⁶ G06F9/44, G06F13/00, G06F15/16

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Jitsuyo Shinan Koho 1979-1998 Toroku Jitsuyo Shinan Koho 1994-1998
 Kokai Jitsuyo Shinan Koho 1971-1994

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	Report of 38th Programing Symposium, January 1997 (Tokyo) Yoshiaki Mima et al., "Moving Agent Frame Work: Aglets (in Japanese)" p.117-124	1-12
A	Research Report of Information processing Society of Japan, January 1997 (Tokyo), Vol. 97, No. 13 Hiyoyuki Tarumi et al., "Work Web System: System for Work Flow Dynamic Replanning (in Japanese)", p.97-102	1-12
A	Hot Topics Presented in the Japanese Society for Artificial Intelligence and Papers of Technical Meetings on Parallel Artificial Intelligence, November 1996 (Tokyo) Vol. 2 Yasuo Nagai et al., "Plangent I: Intelligent Network Agent (in Japanese)" p.29-36	1-12
A	Mobile Agents First International Workshop MA'97 Proceedings, Berlin, Germany, April 8, 1997 (08. 04. 97) J. Baumann et al., "Communication Concepts for Mobile Agent Systems" p.123-135	1-12

☒ Further documents are listed in the continuation of Box C.
 ☐ See patent family annex.

* Special categories of cited documents:	"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A" document defining the general state of the art which is not considered to be of particular relevance	"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E" earlier document but published on or after the international filing date	"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&" document member of the same patent family
"O" document referring to an oral disclosure, use, exhibition or other means	
"P" document published prior to the international filing date but later than the priority date claimed	

 Date of the actual completion of the international search
 March 31, 1998 (31. 03. 98)

 Date of mailing of the international search report
 April 14, 1998 (14. 04. 98)

 Name and mailing address of the ISA/
 Japanese Patent Office

Authorized officer

Facsimile No.

Telephone No.

INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP97/04864

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	Mobile Agents First International Workshop MA'97 Proceedings, Berlin, Germany, April 8, 1997 (08. 04. 97) D.T. Chang and S. Covaci "The OMG Mobile Agent Facility: A Submission" p.98-110	1-12
A	Mobile Agents First International Workshop MA'97 Proceedings, Berlin, Germany, April 8, 1997 (08. 04. 97) H. Peine and T. Stolpmann "The Architecture of the Ara Platform for Mobile Agents" p.50-61	1-12
A	Worldwide Computing and Its Applications International Conference WWCA'97 Proceedings, Tsukuba, Japan, March 1997 D.B. Lange et al., "Aglets: Programming Mobile Agents in Java" p.253-266	1-12
P, A	IEEE Internet Computing, Vol. 1, No. 4 (August 1997) J. Kiriya and D. Zimmerman "A Hands-on Look at Java Mobile Agents" p.21-30	1-12
P, A	IEEE Internet Computing, Vol. 1, No. 4 (August 1997) B. Krulwich "Automating the Internet: Agents as User Surrogates" p.34-37	1-12
P, A	IEEE Internet Computing, Vol. 1, No. 4 (August 1997) J.M. Andreoli, F. Pacull and R. Pareschi "XPECT: A Framework for Electronic Commerce" p.40-48	1-12
P, A	IEEE Internet Computing, Vol. 1, No. 4 (August 1997) A. Ohsuga et al., "Plangent: An Approach to Making Mobile Agents Intelligent" p.50-57	1-12

国際調査報告

国際出願番号 PCT/J P 97/04864

A. 発明の属する分野の分類 (国際特許分類 (IPC))

Int^o G 06 F 9/44

B. 調査を行った分野

調査を行った最小限資料 (国際特許分類 (IPC))

Int^o G 06 F 9/44, G 06 F 13/00, G 06 F 15/16

最小限資料以外の資料で調査を行った分野に含まれるもの

日本国実用新案公報 1979-1998年
 日本国公開実用新案公報 1971-1994年
 日本国登録実用新案公報 1994-1998年

国際調査で使用した電子データベース (データベースの名称、調査に使用した用語)

C. 関連すると認められる文献

引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求の範囲の番号
A	第38回プログラミング・シンポジウム報告集、1月、1997 (東京) 美馬義亮他「移動エージェントフレームワーク： Agentsについて」P.117-124	1-12
A	情報処理学会研究報告、1月、1997年(東京)，第97巻 第13号 垂水浩幸他「ワークウェブシステム： ワークフロー動的再計画の方式」P.97-102	1-12
A	人工知能学会ホットトピックスと並列人工知能研究会資料、 11月、1996年(東京) 第2巻 永井保夫他「Plangent I： インテリジェント・ネットワークエージェント」P.29-36	1-12

☒ C欄の続きにも文献が列挙されている。☐ パテントファミリーに関する別紙を参照。

* 引用文献のカテゴリー

「A」 特に関連のある文献ではなく、一般的技術水準を示すもの
 「E」 先行文献ではあるが、国際出願日以後に公表されたもの
 「L」 優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献 (理由を付す)
 「O」 口頭による開示、使用、展示等に言及する文献
 「P」 国際出願日前で、かつ優先権の主張の基礎となる出願

の日の後に公表された文献

「T」 国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの
 「X」 特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの
 「Y」 特に関連のある文献であって、当該文献と他の1以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの
 「&」 同一パテントファミリー文献

国際調査を完了した日

31.03.98

国際調査報告の発送日

14.04.98

国際調査機関の名称及びあて先

日本国特許庁 (ISA/J P)
 郵便番号100-8915
 東京都千代田区霞が関三丁目4番3号

特許庁審査官 (権限のある職員)

川崎 優

印

5 B

8944

電話番号 03-3581-1101 内線 3546

C (続き) . 関連すると認められる文献		
引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求の範囲の番号
A	Mobile Agents First International Workshop MA'97 Proceedings, Berlin, Germany, 8. 4月. 1997 (08. 04. 97) J. Baumann et al 「Communication Concepts for Mobile Agent Systems」 P.123-135	1-12
A	Mobile Agents First International Workshop MA'97 Proceedings, Berlin, Germany, 8. 4月. 1997 (08. 04. 97) D.T.Chang and S.Covaci 「The OMG Mobile Agent Facility:A Submission」 P.98-110	1-12
A	Mobile Agents First International Workshop MA'97 Proceedings, Berlin, Germany, 8. 4月. 1997 (08. 04. 97) H.Peine and T.Stolpmann 「The Architecture of the Ara Platform for Mobile Agents」 P.50-61	1-12
A	Worldwide Computing and Its Applications International Conference WWCA'97 Proceedings, Tsukuba, Japan, 3月. 1997 D.B.Lange et al 「Aglets:Programming Mobile Agents in Java」 P.253-266	1-12
P、A	IEEE Internet Computing, 第1巻,第4号 (8月. 1997) J.Kiniry and D.Zimmerman 「A Hands-on Look at Java Mobile Agents」 P.21-30	1-12
P、A	IEEE Internet Computing, 第1巻,第4号 (8月. 1997) B.Krulwich 「Automating the Internet:Agents as User Surrogates」 P.34-37	1-12
P、A	IEEE Internet Computing, 第1巻,第4号 (8月. 1997) J.M.Andreoli,F.Pacull and R.Pareschi 「XPECT:A Framework for Electronic Commerce」 P.40-48	1-12
P、A	IEEE Internet Computing, 第1巻,第4号 (8月. 1997) A.Ohsuga et al 「Plangent:An Approach to Making Mobile Agents Intelligent」 P.50-57	1-12